



Europäisches Patentamt  
European Patent Office  
Office européen des brevets

⑪ Publication number:

0 257 581  
A2

⑫

## EUROPEAN PATENT APPLICATION

⑬ Application number: 87112152.1

⑭ Int. Cl.: G06F 15/06

⑮ Date of filing: 21.08.87

⑯ Priority: 29.08.86 US 902343

⑰ Date of publication of application:  
02.03.88 Bulletin 88/09

⑲ Designated Contracting States:  
BE CH DE ES FR GB IT LI NL SE

⑳ Applicant: International Business Machines Corporation  
Old Orchard Road  
Armonk, N.Y. 10504(US)

㉑ Inventor: Li, Hungwen  
60 Deerfield Lane South  
Pleasantville, N.Y. 10570(US)

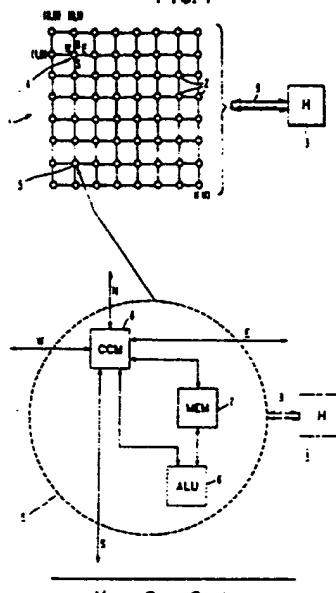
㉒ Representative: Teufel, Fritz, Dipl.-Phys. et al  
IBM Deutschland GmbH. Europäische  
Patentdienste Postfach 265  
D-8000 München 22(DE)

㉓ Polymorphic mesh network image processing system.

㉔ Polymorphic mesh uses physical mesh connection to form twelve useful connection patterns for each of the processing elements (2) making up an image processor of cellular automata under software control. Each processing element includes a limited mesh of interconnections to related processing elements. This provides for programmable choice of network configuration. The limited mesh of network interconnections is controlled by information stored in a register within the affected processing element. The interconnection pattern controlled by this information is invoked by programming, or by the combination of programming and process data, so as to configure the network of processing elements dynamically in the desired mesh. Representative configurations are:

string; mesh; tree; cube; pyramid.

FIG. 1



EP 0 257 581 A2

## POLYMORPHIC MESH NETWORK IMAGE PROCESSING SYSTEM

## BACKGROUND OF THE INVENTION

## 1. Field of the Invention

5

This invention relates to an array processor made up of a network of processing elements, and more particularly relates to an array processing network in which each processing element in the array of processing elements is equipped with a program-accessible connection control mechanism with a controllable limited mesh of interconnections to related processing elements, so as to provide a programmable choice of network configuration,

## 2. Description of the Prior Art

15 The following publications are representative of the prior art:

## Published Articles

- 20 1. Sternberg, "Biomedical image processing," Computer, Jan. 1983, shows an array of cellular automata of identical cells connected to their nearest neighbor for iterative neighborhood processing of digital images. A serpentine shift register serially configures neighborhood inputs to the  $3 \times 3$  neighborhood.
2. Turney et al, "Recognizing Partially Occluded Parts," IEEE Transactions on Pattern Analysis and Machine Intelligence, July, 1985, pp.410-421, shows the use of a variety of techniques including Hough transform, and weighted template matching.
- 25 3. Mudge et al, "Efficiency of Feature Dependent Algorithms for the Parallel Processing of Images," IEEE 0190-3918/83/0000/0369 1983, 369-373, shows how the architecture of an image processing system can benefit from configuration as multiple subimage processors in which processing elements communicate through some form of communication network. The authors explore the difference between feature-dependent algorithms and feature-independent algorithms.
- 30 4. Sternberg et al, "Industrial Morphology," shows the combination in a single system of image processing and of pattern recognition.
5. D.E. Shaw, "The NON-VON Supercomputer," internal report, Columbia University, Aug. 1982, shows a massively parallel system with an I/O switch in each processing element and with flag registers to activate and deactivate individual PEs.
- 35 6. M.J. Kimmel, R.S. Jaffe, J.R. Mandeville, and M.A. Lavin, "MITE: Morphic Image Transform Engine, An Architecture for Reconfigurable Pipelines of Neighborhood Processors, IBM RC11438, Oct. 10, 1985, shows a reconfigurable network of processing elements capable of a variety of interconnections of Pe to PE via bus connections under operator control.
- 40 7. A.J. Kessler and J.H. Patel, "Reconfigurable Parallel Pipelines for Fault Tolerance," IEEE, CH1813-5/82/0000/0118, 1982, shows reconfigurable pipeline connection for graceful degradation.
8. S. R. Sternberg, "Parallel Architecture for Image Processing," IEEE, CH1515-6/79/0000-0712, 1979, shows a PE network with full connectivity.
- 45 9. T. N. Mudge, E. J. Delp, L. J. Siegel and H. J. Siegel, "Image Coding Using the Multimicroprocessor System PASM," IEEE, 82CH1761-6/82/0000/0200, 1982, shows processing element interconnection by an interconnection network.
10. S. R. Sternberg, "Language and Architecture for Parallel Image Processing," Pattern Recognition in Practice, North-Holland Publishing Co., 1980, shows a complex network of PEs and explains operation.

50

- Patents • U. S. Patent 4,174,514, November 13, 1979, shows an array processor with adjacent neighborhood processing elements interconnected for mutual access to the image data in overlap areas of two adjacent image slices.
- U. S. Patent 4,215,401, CELLULAR DIGITAL ARRAY PROCESSOR, July 29, 1980, shows an array processor in which each processing element "cell" includes two accumulators arranged to connect that cell

- (except those at the array edge) with its two neighboring cells along one axis and with its two neighboring cells along the orthogonal axis.
- U. S. Patent 4,380,046, Fung, MASSIVELY PARALLEL PROCESSOR COMPUTER, April 12, 1983, shows an image processor which performs spatial translation by shifting or "sliding" of bits vertically or horizontally to neighboring processing elements, P register to P register, as permitted by another register, called G register. Each processing element includes an arithmetic, logic and routing unit (ALRU), an I/O unit and a local memory unit (LMU). The ALRU constitutes three functional components, a binary counter/shift register subunit, a logic-slider subunit (P register) and a mask subunit (G register).
  - U. S. Patent 4,398,176, Dargel et al, DATA ANALYZER WITH COMMON DATA/INSTRUCTION BUS, August 9, 1983, shows an image processing array in which each processing element includes external command control lines to control whether bus information is to be used as instruction or as data. Each processing element also includes mechanism to determine from the bit structure of an instruction whether the instruction is local or global, and to stop forward transmission if local.
  - U. S. Patent 4,601,055, Kent, IMAGE PROCESSOR, July 15, 1986, shows an iconic-to-iconic low level image processor with pixel-by-pixel forward transformation and with pixel-by-pixel retrograde transformation.

The prior art provides for permanent configuration of processing elements in a pipeline or other pattern image processing system. The prior art provides for reconfiguration of an image processing system via switching networks and busses. The prior art provides for convenient bit transfer to adjacent processing elements. The prior art does not however, teach nor suggest the invention, which provides for very high speed switching within the processing element, programmable as a polymorphic mesh, so as to optimize dynamically the configuration to the data being processed in a configuration such as: string; mesh; tree; cube; pyramid.

Cellular automata has been found very useful for image processing, computer vision and other computations in physics. All existing interconnection networks for cellular automata assume a fixed pattern such as a string, a mesh, a tree, a cube or a pyramid, etc.. Each pattern is good for certain types of computing but is poor for computations that do not match the pattern. Since the network interconnection pattern is built in, it is fixed; it can not be changed even when a mismatch is detected. The mismatch leads to poor efficiency.

For example, an NxN mesh is an optimal interconnection for local operations in image processing, but its performance is poor in computing a global operation (e.g. it takes N cycles to compute MINIMUM, a linear complexity). On the other hand, a tree interconnection is optimal for computing MINIMUM (it takes only log N cycles, a logarithmic complexity) but is very inefficient in computing the local operations of an image because of the lack of the neighborhood connection.

Besides the general inefficiency in computing when the interconnection does not match the algorithm, the fixed-pattern approach is inflexible in designing an algorithm. This is mainly caused by the restriction in data flow. For example, in the string interconnection, the data flow is one direction only, from left to right. Such a restriction confines the algorithm domain; only the algorithms that have a "string" data flow can benefit from the "string" network. In this regard, the fixed networks are very special-purpose and have a very narrow range of applications.

Another disadvantage of the fixed interconnection pattern is that it does not support efficiently iconic and intermediate processing simultaneously. Such disadvantage is specific to one important application of the cellular automata, computer vision, in which both iconic (or image) processing and the transformation from iconic to symbolic information (called intermediate level processing) are two integral parts. A serious implication of this disadvantage is the I/O problem, because the image after iconic processing needs to be shipped outside of the network for further intermediate processing.

## 50 SUMMARY OF THE INVENTION

The object of the invention is to provide program-accessible convenient high speed connectivity to each processing element in an array, so that processing elements may be programmably grouped in an effective manner without incurring the cost of the complex connections required for universal connectivity.

Another object of the invention is to provide convenient programmable control of connectivity of each processing element in the array, so that processing elements may be programmably regrouped from time to time, both under adaptive control by the computer as it senses the need for optimization regrouping and under operator control as the operator foresees the need for optimization regrouping.

Another object of the invention is to provide an external memory data connection to the processing element, by which certain hardware may be eliminated and additional flexibility of operation may be gained.

5 A feature of the invention is the use of an array of polymorphic mesh processing elements, each of which has processing capability embodied in an arithmetic and logic unit with memory, and also has programmable connection control capability with geographic destination connections and also logical connections.

10 Another feature of the invention is a provision for programmable short-circuit capability in the polymorphic mesh processing element. The short-circuit capability allows a series of intervening processing elements to serve simply as wire equivalents in transmitting data from a sending PE to a remote PE without cycle delay.

15 Another feature of the invention is a "polymorphic-mesh network," which is a composite network of a conventional mesh external to each PE and an internal network within each PE, to accommodate standard patterns and other new useful patterns through software control so that the connection can be matched to the computing. Through the "polymorphic" feature, the network can "reshape" itself adaptively, to allow flexible algorithm design and to cover wider application spectra.

20 A specific feature, related to computer vision, supports the intermediate level processing (iconic to symbolic transformation) by the polymorphic mesh, resulting in an efficient architecture while avoiding the serious I/O problem.

25 Another specific feature of the invention is the provision of flag registers in the processing elements for use in conditional operations including reconfiguration for adaptive self-optimization and for fail-soft capability.

30 Another feature of the invention is the provision of a limited number of multi-bit pattern registers, each accessing a limited number of software selectable hardwired patterns, useful in cellular automata, which can be formed by the polymorphic mesh. These patterns include bus, several trees, cube and pyramid, each of which is optimal for a related type of computation, selectable by a crossbar switch in response to a bit pattern in a selected one of the pattern registers.

35 An advantage of the invention is its high throughput speed at relatively low cost, achieved by providing programmable limited connectivity within the processing element, so as to permit optimization of array connection of processing elements both physically and electronically.

40 Another advantage is that results of intermediate level processing may be used in conjunction with programming to provide adaptive reconfiguration efficiently as a joint function of programming and intermediate results of processing. This means that intermediate data can be used to invoke an adaptive optimization regrouping function; adaptive self-optimization and fail-soft capabilities result.

45 Another advantage is the simplicity and two-dimensional aspect of the invention, which permits vast numbers of interconnectable processing elements to be arrayed on a small number of chips.

The foregoing and other objects, features and advantages of the invention will be apparent from the more particular description of the preferred embodiment of the invention, as illustrated in the accompanying drawings.

#### 40 BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a system block diagram of an image processor made up of a network of polymorphic mesh processing elements, with one processing element shown in functional block diagram form.

45 FIG. 2 is a diagram of the switching capability of the connection control mechanism CCM of a polymorphic mesh processing element.

FIG. 3 is a functional block diagram of the connection control mechanism of a polymorphic mesh processing element.

50 FIG. 4 is a diagram showing formation of linear string arrays from polymorphic mesh processing elements.

FIG. 5 is a diagram showing formation of row trees from polymorphic mesh processing elements.

FIG. 6 is a diagram showing an alternative view of row trees.

FIGs 7-10 are simplified diagrams showing chip area comparison and communication distance improvement as a result of using polymorphic mesh processing elements.

55 FIG. 11 is a diagram showing formulation of reverse row trees from polymorphic mesh processing elements.

FIGs. 12-20 are diagrams showing representative choices of array configuration conveniently achievable by programmable interconnection of polymorphic mesh processing elements.

FIG. 21 is a more detailed block diagram of an individual polymorphic mesh processing element according to a preferred embodiment of the Invention.

## 5 DESCRIPTION OF A PREFERRED EMBODIMENT OF THE INVENTION

FIG. 1 shows a polymorphic mesh network image processing system made up of an MxM array 1 of processing elements 2 under control of host computer H 3. Each processing element has a limited set of connections; in the preferred embodiment there are four connections, one connection to each of its adjacent orthogonal (non-diagonal) neighbors. These orthogonal neighbors will be referred to as Cartesian neighbors. These orthogonal connections are identified for processing element 4 with the directional identifications NESW. The function of these connections is to present the output of any processing element directly to a limited number of its neighbors (four in the preferred embodiment). Overall programming control and housekeeping control is by host computer 3, via bus 9.

15 One processing element is shown in greater detail. Processing element 5 is shown expanded to provide drawing space for internal organs ALU 6, MEM 7 and CCM 8 and NESW connections.

Each processing element (PE) in the preferred embodiment is equipped with four Cartesian connections. These Cartesian connections are designated NESW for convenience in discussion. They connect to the respectively adjacent PEs in the designated directions. This simple mesh of Cartesian connections 20 would be capable, without CCM 8, of making connection to the adjacent PE, which is a very important capability in image processing. This simple mesh is susceptible to convenient very large scale integration (VLSI) manufacturing.

25 Non-Cartesian PEs are not wired for direct connection. Diagonal connection and remote connection are not available on wires or metallization. Such connections would make manufacturing much more difficult; bundles of wires would provide bulk distance with its inherent speed-of-light delay.

In the preferred embodiment, non-Cartesian PEs are accessed via intervening Cartesian PE's through programmed control of their respective CCM 8 capability. The CCM 8 pattern of connection is such that the input is effectively short-circuited to the desired output. Connection may follow the pattern of the chess rook (straight Cartesian with optional extension) or the chess knight with optional extension (Cartesian X, 30 Cartesian Y == remote off-Cartesian) but not the chess bishop (diagonal with optional extension). Complex routing patterns may be set up. Complex routing to a non-Cartesian destination may be set up to pass through a great number of PEs without cycle delay.

As an alternative to the simple Cartesian connections of the preferred embodiment, the limited set of 35 PE-PE interconnections might also include connections to diagonally adjacent PEs. Direct connections to highly remote PEs, however, are prohibitively complex, considering that such connections can be made under program control according to this invention by combinations of Cartesian or other simple connections.

FIG.2 is a diagram of the switching capability of the connection control mechanism CCM 8 as shown in FIG. 1. The essential function is as a switching network to connect any one of the connections NESW in the X crossbar with any one of the connections NESW in the Y crossbar, under control of the bit values in a 40 pattern register. The preferred embodiment provides for selection between two pattern registers, with the selection of pattern register controlled by a bit value in a pattern selection register. Note that FIG. 2 is diagrammatic and does not show details of actual hardware. In the preferred embodiment not all of the connections available in a 4x4 matrix are necessarily used although all are indeed available. Matrix 10 is shown as having inputs 11 in the Y dimension with connections to conductor S12 in the X dimension. As 45 shown, this is set up to connect N connector 13 via connection electronics 14 and 15 and intersection 16 to E connection 17. Different connections to SWN connections 18, 19 and 20 may be alternatively or simultaneously activated. The control is by pattern register 21 bit values or by pattern register 22 bit values. Pattern register selection is by pattern register selection register 23. Pattern registers 21 and 22 are sixteen bits each as shown by the slash on the connection line with the 16. Connections to connection lines 24, 50 25 and 26 are also available. Crossbar switch 10 makes any one or plurality of sixteen connections as controlled by the bit values in the selected one of pattern registers 21 or 22.

Each processing element connects to one or more of its neighbors as directed by the bit values in one 55 of its two pattern registers. Instantaneous change of connection can be made simply by switching control to the alternate pattern register. Selection of pattern registers is by the bit value in a simple binary pattern selection register. For purposes of explanation, the pattern registers may be considered standard pattern register and alternate pattern register, respectively accessed by 1 or by 0 value in the binary pattern selection register.

FIG. 3 is a functional block diagram of CCM 8 FIG. 1, used to implement the switching capability described in FIG. 2 and also to carry out certain other direction and logical function controls. A simplified logical connective capability 30 is shown for AND, OR, XOR (Exclusive OR), and ANDALLBIT capability. Other capabilities are also available but as the complexity increases the cost increases. The connection

- 5 control mechanism receives flag inputs which are stored in first flag register F1 31 and second flag register F2 32. These flags may be passed forward and may also be used to alter the control. Shift register mask 33 is provided to present outputs SRM 34 TRUE and SRM 35 COMPLEMENT, both of which are used to select a limited subset of the SRX 37 and SRY 39 bit values to control logical connective box 30.

The X register 36 and SRX shifter 37 function to control the logical connective and also the geometrical connection to a neighboring processing element. Similarly, Y register 38 and shift register Y SRY 39 function to control geometric and logical selection in the Y dimension. Usually X register 36 and Y register 38 contain the Cartesian coordinates of a processing element in the system. Shifters SRX 37 and SRY 39, in conjunction with SRM 33, are used to derive any contiguous bit group of X and Y. Detailed functions of FIG. 3 are outlined by twelve examples, formulating twelve different patterns from the polymorphic-mesh.

- 10 15 Each processing element is set up to carry out an arithmetical logical operation or logical transform operation on values presented to it, or, alternatively, to perform a no-op. In addition to the operation or no-op, the PE is set to connect with selected neighboring processing elements. One such connection to neighbor processing elements is to serve as a short circuit. The short circuit connection is essentially instantaneous. Connection occurs at the speed of electricity (speed of light) rather than with a one-cycle delay as is common with ordinary operations, including no-op. It is thus possible to bypass several processing elements in order to make the desired connection to a processing element which is a non-adjacent Cartesian neighbor. It is possible to make a zigzag move, to make connection to a remote processing element which is neither in the same column nor in the same row. In this fashion, even though the array connections are without diagonals, connection can be made to adjacent diagonal or remote 20 25 diagonal or off-diagonal remote processing elements.

While additional explanation will be made, it should be understood at this point that the polymorphic mesh image processing system, properly program controlled, can be configured to carry out a processing effort involving a complex interconnection of processing elements. Each processing element does its appropriate arithmetic or logical operation or transform operation or no-op, with respect to information provided to it, or the processing element may be short-circuited so as to be bypassed without cycle delay.

A certain amount of sophistication in connections and in connection logic might be available as a function of the number of bit values in the pattern registers, and as a function of the number of pattern registers, of the connection control mechanism. But such sophistication would be costly, because of the large number of replications required by the large number of processing elements. In order to keep costs of 30 35 the processing element down, costs being measured in terms not only of money but of complexity and path length, the preferred embodiment has a limited repertoire of optimized connections. This repertoire is depicted in FIGS 4 through 20.

FIG. 4 shows the formulation of linear arrays. There is a west to east linear array 41, and there is a north to south linear array 42. FIGs 5 and 6 show two techniques for forming row trees. In the first 40 technique, X strings 45, 46, 47 and 48 through 51 are shown in FIG. 5. These X strings of row trees are of different lengths. FIG. 6 shows a tree in its more traditional presentation as tree connection 52. In this traditional tree configuration seven inputs filter out to a single processing element in four steps.

FIG. 7-10 show a chip area comparison and communication distance improvement as a result of using polymorphic mesh processing elements. The chip area for polymorphic mesh 61 as shown in FIG. 7 is quite compact as contrasted to pattern 62 of an orthogonal tree, FIG. 9. The average communication distance in a 3x3 window, as shown in FIG. 8 pattern 63, is 1.5 using the polymorphic mesh. The same 3x3 window 64 in FIG. 10 has an average communication distance of 2.125.

FIG. 11 is a diagram showing formulation of reverse row trees from polymorphic mesh processing elements. This is very similar to row trees shown in FIG. 5. While a row tree is used to collect information 50 from its leaves, a reverse row tree is to distribute information to its leaves.

FIGS. 12-20 show selected choices of array configuration conveniently achievable by programmable interconnection of polymorphic mesh processing elements. Certain ones of the processing elements in the figures are shown as squared circles, to indicate processing operations, as contrasted to intervening processing elements used simply for pass-through, which are shown as simple circles. Note that each PE 55 can perform both the pass-through function and a processing operation, as programmed. Each pattern is reduced to a set of sixteen bit values and presented via the appropriate pattern register.

FIG. 21 is a detail of the preferred embodiment polymorphic mesh processing element according to the invention. Note that much of the mechanism shown is relatively standard. This relatively standard hardware is shown outside the broken line box 93. The standard hardware features ALU 94 which provides outputs 1 and 2 as well as appropriate input multiplexes. Inputs may come via instruction terminal 94 and also from registers NSE and W 95 which in turn are fed from memory 1 M1, memory 2 M2, ALU output 1 or output 2. The local memory at 101 is available for appropriate purposes of computation and housekeeping related to the ALU. The local memory 101 can be extended to the external memory whose content is fed into the processing element via an external memory data wire EMD 102. The local and external memory content are multiplexed via multiplexer 103 and connected to M1 or/and M2. Output signals out 1 and out 2 are fed back to the same processing element 93 and may be also fed to other processing element as selected by the CCM.

The EMD connection and plurality of buses M1 and M2 provide for an external memory data connection to the internal memory means, processing means and connection control mechanism, whereby the processing element may be operating with local memory and external memory simultaneously.

The connection at EMD 102 is important, in that it permits direct connection of the individual PE to an external memory which may be in the host H 3 or may be a standalone memory not shown. This EMD connection, not present in ordinary processing element in array processors, makes it possible to provide the equivalent of FIG. 3 from an external memory. The EMD connection also makes possible a supplement to the hardware of FIG. 3, as well as great flexibility of operation and setup of polymorphic-mesh processing element networks.

The switching capability implemented in FIG. 3 via its directional and logic function control is related to external memory and EMD 102. A wide spectrum of implementation means for FIG. 3 is possible, ranging from having all functions illustrated in FIG. 3 resident in each and every processing element to having all these functions resident external to all PEs but delivering the resultant condition via connection EMD 102 to a pattern register selection register Rp 99.

Two pattern registers, PR0 97 and PR1 98, allow instantaneous switching from one connection pattern to the other without loss of any instruction cycle. The instantaneous switching is controlled by a one-bit register, the pattern register selection register (Rp 99). When it is determined that the values in one of the two pattern registers are no longer necessary for use, that pattern register can be loaded with a new pattern. Such loading can be free, that is, can be carried out at the same time as the concurrent ALU operation. It must be remembered that the processing elements in image processing systems are normally one-bit processing elements, and in any case are relatively simple. It is a relatively significant effort to load the pattern registers 97 and 98, which are 16 bits each. Pattern register selection register 99 must also be loaded, but this is a single bit.

There are occasions when it is necessary to close down processing to reload pattern registers 97 and 98. If this occurs it would normally take 32 cycles to load the pattern registers and a 33rd cycle to load the pattern register selection register 99. In many cases, however, there need be no time used specifically for loading the pattern registers. In the case of appropriate instructions, known to be appropriate, the operator can load one bit into the pattern register 97 or one bit into the pattern register 98 or one bit into the pattern register selection register 99. During the cycle in which ALU 96 is carrying out an arithmetic or logical operation over a period of time during which the arithmetic and logical unit 96 is operating at full capacity, it thus maybe possible to reload one or all of the registers in CCM. Such freeloading is by a path from instruction in set 94 via multiplexes in ALU 96 and feedback from out 1 or out 2, assuming appropriate setup of instruction gate 100 to carry out the loading function.

Note that the selected sixteen bit values from pattern registers 97 or 98 are to control the detailed setting of the 4x4 crossbar switch 104. Its detail can be referred to FIG. 2.

In a typical operation the processing element will be either short circuited or active. If short circuited, a pattern register such as register 97, set for short circuit connection, takes over and carries out the short circuit connection via that processing element to one or more other processing elements. In the case when the processing element is active pattern register selection register 99 would be set for action and would switch control from standard pattern register 97, set for bypass via short circuit, to alternate pattern register 98, set for directing inputs and outputs for the activity.

These activities will be explained in the following paragraphs.

The polymorphic mesh, shown in FIGS. 1-3, is capable of a number of patterns limited by the complexity of its connection control mechanism CCM. The CCM pattern repertoire, being the union of these patterns, is optimal for the union of the computing types covered separately.

One control algorithm is described in the invention for each pattern respectively. All control algorithms are simple to implement and most importantly use the same set of hardware to generate the desired pattern on-line.

- 5 A hardware mechanism is depicted in the invention to carry out the formation of all patterns in a systematical and consistent way. The mechanism is simple to implement and very suitable for VLSI implementation.

As a most distinguished feature of the polymorphic-mesh, many algorithms of linear complexity ( $O(N)$ ) are reduced to logarithm complexity ( $O(\log N)$ ), a theoretical optimal. Therefore the  $N/\log N$  speedup is gained by the architectural novelty; for a network of  $1024 \times 1024$ , the speedup is 100.

- 10 Specific to computer vision, after the iconic processing by one pattern (mesh), the resulting image is not shipped outside of the network. Rather, it is further processed by another pattern (e.g. tree) to transform the iconic information to symbolic information (e.g. how many pixels whose values are greater than 133?). Because of the polymorphic capability, the data do not have to be output, consequently the I/O rate is significantly reduced (e.g. five orders of magnitude reduction for the "how many" example in a  $1024 \times 1024$  image). The saving from the I/O reduction contributes to the speedup on top of the speedup due to computing in a compound manner.

- 15 Another pattern, Diagonal-Span-tree, can be formed by the polymorphic-mesh to facilitate the computing of  $Ax + By + C$  in logarithm time where A, B and C are constant and  $(x, y)$  is the coordinate of a pixel. This capability is useful in both computer graphics and computer vision. For computer graphics, it is useful  
20 in display convex polygon, in creating shadow, in clipping, in drawing spheres, in computing adaptive histogram equalization, in texture mapping and anti-aliasing. For computer vision, such a capability is useful in generating a line mask, a band mask and a polygonal mask. It is also useful in computing Fast Hough Transform and its inverse for detecting lines in a noisy image and other applications.

- 25 The preferred embodiment includes a hardware mechanism to generate twelve useful cellular automata patterns and twelve control algorithms, one for each pattern, to reshape the polymorphic-mesh into the corresponding pattern under software control. The following sections describe the polymorphic hardware mechanism and the control algorithms.

- 30 One noticeable feature of polymorphic mesh is the capability to "reshape" itself adaptive to the condition of the processing. As described previously, the pattern registers 97 and 98 can be loaded concurrently with the ALU operation; as a result, it can be considered that each PE has P patterns  
35 ( $P_1, \dots, P_p$ ) at its disposition.

- 35 The reshaping that is adaptive to the condition C of the processing allows each PE to assume a pattern  $P_i$  as a function C. Initially, all PEs start from a pattern, say  $P_1$ , and processing begins under the initial pattern. As processing goes, each PE senses its local condition (this can be a test of convergence of a value, for example) and decides whether to stay in the current or replace it with a new pattern.

A condition can be global, meaning that the host can sense each PE condition collectively, then decide the choice of pattern and feed back to all PEs. A condition can also be local. The new pattern may be fed to an individual PE, or to a group of PEs as appropriate.

- 40 Choice of a new pattern can be made in several ways.

(1) prescheduled:

- 45 a sequence of patterns (e.g.  $P_1 P_2 \dots P_p$ ) are scheduled and will be adopted in such a priority. When the need of a new pattern is detected, (while using  $P_1$ ) the next pattern  $P(i + 1)$  will be used. Then  $P(i + 2)$  will be loaded when possible to the unused pattern register concurrent with the operation.

50 (2) function of C:

55

An example to show the adaptive reshaping is as follows:

A pattern of 3x3 window is established as Pi to allow filtering operation and an index is identified as condition C to measure the effectiveness of the filtering. The goal here is to increase the window size when C is not satisfactory. In this regard, we establish P2 as a 5x5 window, P3 as 7x7, P4 as 9x9 etc.

- Another example of adaptive reshaping is the "soft fail" application. It is common to design a condition 5 C in each PE such that a malfunction reflected. Such a condition then can be used to decide a new suitable connection pattern Pi.

#### POLYMORPHIC-MESH MECHANISM

10

The polymorphic mesh (Figure 1) consists of an array 1 of MxM processing elements (PE). Each PE has four physical wires communicating with four non-diagonal neighbors, one for each neighbor. These communicating wires are denoted as E, W, S and N as shown for PE 4. PE 5 is shown expanded to show arithmetic and logical unit (ALU) 6, memory (M) 7, and connection control mechanism (CCM) 8, ALU 6 15 and 7 are relatively standard features in image processor processing elements. The connection control mechanism is not standard, but rather is in the special configuration according to this invention.

As shown in FIGS 1 and 2, each PE consists of three functional blocks: the connection control mechanism 8, the memory block and the Arithmetic Logic Unit (ALU) block 6. The connection control mechanism 8 takes four wires (E, W, N and S) ALU output and memory (M) as inputs and reroute them as 20 outputs. The routing is accomplished by "SHORT\_CIRCUITing" any input A (for example, 10 in in FIG. 2) to any output B. The signal appears on wire A is logically as that of wire B where A and B can be any of the inputs to the connection control unit. For example, "SHORT\_WE" will logically equalize wire W 24 and wire E 26.

FIG. 3 shows the functional units of CCM 8. The action "SHORT\_CIRCUIT" is conditional and the 25 conditions are created by the connection control mechanism shown in Figure 3. Each PE has two 1-bit flags F1 31 and F2 32 for generating condition signals. Each PE is equipped with one shift register SRM 33 which can shift in both direction logically or arithmetically and supply true and complement outputs 34 and 35.

Each PE contains a pair of registers X 36 and Y 37 where register X holds the PE's row position (0 ≤ X 30 ≤ M-1) and register Y holds the PE's column position (0 ≤ Y ≤ M-1). Both register X and Y have a shift register SRX 38 and SRY 39, each of which can be loaded the content of X and Y respectively and can shift in both directions logically or arithmetically. The bit shifted out of SRX is BSRX and that of SRY is BSRY.

Several functions are provided to generate the condition on which the "SHORT\_CIRCUIT" action is based.

LOAD reg value: this function will load the "value" into SRM, F1 or F2 via instruction or memory;

COPYSR reg: this function will copy reg X to reg SRX or Y to SRY or both

AND/OR/XOR reg: this function performs one of the following,

"AND/OR/XOR" X with SRX and X with SRM (or inverted SRM);

"AND/OR/XOR" Y with SRY and Y with SRM (or inverted SRM);

both of above;

ANDALLBIT reg: this function performs "AND" on all bits of reg; it produces one condition bit XANDALL if reg is X, or one condition YANDALL if reg is Y, or both condition bits.

The "SHORT\_CIRCUIT" action is then based on the combination of BSRX, BSRY, XANDALL, YANDALL, F1 and F2.

45 The remaining two functional blocks of the polymorphic-mesh are rather similar to the conventional design. The memory block can be viewed as a storage that can deliver one bit to the connection control block and/or accept one bit from it per machine cycle. The ALU, although is similar to the conventional design, features on its "conditional" response to the combination of bits BSRX, BSRY, XANDALL and YANDALL in choosing a "SEND" or a "RECEIVE" action along with the "SHORT\_CIRCUIT" action.

50 With the polymorphic-mesh mechanism, twelve patterns formed by the polymorphic-mesh are described in the following. Their corresponding control algorithms are described in order.

## CONTROL ALGORITHMS

## (P1) Linear Array

- 5 As shown in Figure 4, a row linear array of length  $M \times M$  and a column linear array of the same length can be formed from an  $M \times M$  polymorphic mesh by connecting S of PE ( $M-1, i$ ) to N of PE ( $0, i+1$ ) and connecting E of PE ( $i, M-1$ ) to W of PE ( $i+1, 0$ ). The N of PE ( $0, 0$ ) and S of PE ( $M-1, M-1$ ) are the beginning and the end of the column linear array respectively while the W of PE ( $0, 0$ ) and E of PE ( $M-1, M-1$ ) are the beginning and the end of the row linear array.
- 10 The control algorithm is as follows. At per PE cycle,

## LINEAR ()

```

15   {
20     MEM = W; /* action 1*/
25     E = MEM; /* action 2*/
30     MEM = N; /* action 3*/
35     S = MEM; /* action 4*/
40   }
45
50
55

```

- Action (1) takes the datum on W into MEM at the end of the cycle while action (2) puts the content of MEM (at the beginning of the cycle) on E. In combination, actions (1) and (2) create the row array. The data injected through W of PE ( $0, 0$ ) will march eastbound and after  $M$  cycles, the first row will be filled. After another  $M$  cycles, the data in the first row will march to the second row while the new data will fill the first row. Actions (3) and (4) create a column linear array in a similar fashion.

The formulation of the linear array is unconditional. All PEs are taking the same action. The mechanism in the connection control mechanism is not utilized.

35

40

45

50

55

## (P2) Row Tree

M trees (one per row) can be formed from the polymorphic mesh by the following control algorithm.

5

10

## ROW-TREE ()

{

15

int t; /\* t is time step\*/

int pid0; /\* column position of a PE, Process ID0\*/

20

int M, logM; /\*M is the side size of the mesh and logM=log M\*/

int treemask=1; /\* a flag to construct the tree\*/

25

for (t=0; t&lt; logM; t++) {

if(~treemask)

30

{SHORT \_WE; DISABLE;}

if(treemask &amp;&amp; ~pid0&lt;t&gt;)

E = MEM;

35

if(treemask &amp;&amp; pid0&lt;t&gt;)

MEM = W;

treemask = treemask &amp; pid0&lt;t&gt;;

40

}

}

45

Figure 5 illustrates the above control algorithm for a 8-PE case. At t=0, the "treemask"s for all PEs are 1, therefore, every PE is enabled and the "even PEs" send data to "odd PEs". This is indicated by an arrow between every pair of "odd/even" PEs. This also forms the bottom level of the tree.

50

At t=1, by investigating treemask, only the PEs with pid0<0> = 1 (lowest bit of pid0) are enabled. The disabled PEs are not shown by circle but they do establish the connection between PE 1 and 3, and PE 5 and 7 by the action SHORT \_WE. This forms the second level of the tree.

The highest level of the tree is formed by the control at t=2. At this step, only PE 3 and PE 7 are enabled; the rest PEs establish the connection by "SHORT" the W-E path but do not perform any operation (or equivalently do not change any state of MEM).

55

An alternative view of this control algorithm is shown in Figure 6 with the nodes at each level of the tree marked by the processor identification (pid) of the PEs.

The tree pattern is extremely useful in the paradigm of divide-and-conquer. Dedicated tree-machines have been built for special-purpose computing. The complexity of the algorithms in this paradigm is usually  $O(\log N)$  where  $N$  is the size of the input data. The same algorithms usually need  $O(N)$  execution time in a mesh; this represents a 1024:10 speedup for  $N = 1024$ . Important algorithms in this category include MAX, MIN, k-th largest, median, some/none etc.

Using the mechanism in the connection control block, the control algorithm uses register X for pid0, register SRX and flag F1 for "treemask". The content of X register is copied to the SRX so that pid $<t>$  bit is shifted out at time step t in BSRX then ANDed with "treemask" to derive the final condition.

70

*(P3) Column Tree*

Similar to Row Tree, M column trees can be formed by the polymorphic mesh with the row position of the PEs, pid1, as the control and N-S path as the connection. The control algorithm is as follows.

75

```
COLUMN-TREE () {
    int t; /* t is time step*/
    int pid1; /* row position of a PE*/
    int M, logM; /*M is the side size of the mesh and logM=log M*/
    int treemask=1; /* a flag to construct the tree*/

    for (t=0; t< logM; t++) {
        if(~treemask)
            {SHORT_NS; DISABLE;}
        if(treemask && ~pid1<t>)
            S = MEM;
        if(treemask && pid0<t>)
            MEM = N;
        treemask = treemask & pid0<t>;
    }
}
```

Column Trees are useful in transporting the data distributed column-wise in the mesh and converting algorithms with linear complexity ( $O(N)$ ) to logarithmic complexity ( $O(\log N)$ ) as discussed in Row Tree section.

Similar to ROW\_TREE, the COLUMN\_TREE control algorithm uses register Y for pid1, SRY to copy Y, and F2 to hold "treemask". The condition to SHORT\_NS is based on the ANDing of F2 and BSY which produces pid1 $<t>$  at time step t.

## (P4) Orthogonal Tree

Orthogonal Tree (FIG. 9) is a useful network for sorting, matrix operations, minimum spanning tree, FFT and other graph algorithms. It can be formed from the polymorphic mesh by combining the Row Trees and  
 5 the Column Trees by the ORTH\_TREE control algorithm below.

10

```

ORTH_TREE () {
  int t; /* t is time step */
  int pid0; /* column position of a PE, Process ID0 */
  int pid1; /* row position of a PE */
  int M, logM; /*M is the side size of the mesh and logM=log M*/
  int hmask=1, vmask=1; /* flags to construct the tree */

  for (t=0; t< logM; t++) {
    /*cycle 1*/
    if(~hmask)
      {SHORT_WE; DISABLE;}
    if(hmask && ~pid0<t>)
      E = MEM;
    if(hmask && pid0<t>)
      MEM = W;
    hmask = hmask & pid0<t>;
    /*cycle 2*/
  }
}

```

50

55

```

if(~vmask)

{SHORT_NS; DISABLE;}

5      if(vmask && ~pid1<t>)

S = MEM;

10     if(vmask && pid1<t>)

MEM = N;

15     vmask = vmask & pid1<t>;
}

}

```

20 Key advantages of forming the Orthogonal Tree from the polymorphic mesh are  
(1) reduction of chip area: the chip area required to layout the mesh and the orthogonal tree are  $O(N^2)$  and  $O((N^2)(\log N)^2)$  respectively where  $N$  is the side size of the mesh and the number of leafs of the orthogonal tree. This represent a saving at a factor of  $(\log N)^2$ . For  $N=1024$ , the chip area used by the  
25 polymorphic mesh is 1/100 of the orthogonal tree.  
(2) efficient neighborhood operations: PEs in Orthogonal Tree does not connected to its geographical nearest neighbors hence for image processing, many important neighborhood operations can not be performed efficiently because there are no direct communication paths. In fact, more than half of the data in a 3x3 window must be passed up one level in the tree then passed down to the center of the window; the  
30 average distance among data in a 3x3 window is 2.125 as against 1.5 in the polymorphic mesh. A sample 3x3 window for the orthogonal tree is shown in Figure 8 and 10 and the number in the circle is the distance between that datum and the centre of the window.

As summarized in Figure 7 for  $N=4$ , between the polymorphic mesh and the orthogonal tree, the ratio of chip area is 16:46 and that of average distance in a 3x3 window is 1.5:2.1 (Figure 8 and 10).

35 The control algorithm ORTH\_TREE uses register X for pid0, SRX to copy pid0, F1 for hmask, and produces pid0<t> at time step t in BSRX. Symmetrically, the control algorithm uses register Y for pid1, SRY to copy pid1, F2 for vmask, and produces pid1<t> at time step t in BSRY. The conditional SHORT\_WE and SHORT\_NS are based on BSRX, BSRY, F1 and F2.

#### 40 (P5) Reverse-Row Tree

RR-Tree is a top-down tree (as against row trees which are bottom-up) which can be formed from the top level to the bottom level of the tree, a reverse process of the row tree. The control algorithm is shown  
45 below.

50

55

## RR-TREE ()

```

5      {
10     int t; /* t is time step*/
15     int pid0; /* column position of a PE*/
20     int M, logM; /*M is the side size of the mesh and logM=log M*/
25     int treemask=M%2; /* a flag to construct the tree*/
30     int mask; /*an intermediate condition*/
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300
305
310
315
320
325
330
335
340
345
350
355
360
365
370
375
380
385
390
395
400
405
410
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
505
510
515
520
525
530
535
540
545
550
555
560
565
570
575
580
585
590
595
600
605
610
615
620
625
630
635
640
645
650
655
660
665
670
675
680
685
690
695
700
705
710
715
720
725
730
735
740
745
750
755
760
765
770
775
780
785
790
795
800
805
810
815
820
825
830
835
840
845
850
855
860
865
870
875
880
885
890
895
900
905
910
915
920
925
930
935
940
945
950
955
960
965
970
975
980
985
990
995
1000
1005
1010
1015
1020
1025
1030
1035
1040
1045
1050
1055
1060
1065
1070
1075
1080
1085
1090
1095
1100
1105
1110
1115
1120
1125
1130
1135
1140
1145
1150
1155
1160
1165
1170
1175
1180
1185
1190
1195
1200
1205
1210
1215
1220
1225
1230
1235
1240
1245
1250
1255
1260
1265
1270
1275
1280
1285
1290
1295
1300
1305
1310
1315
1320
1325
1330
1335
1340
1345
1350
1355
1360
1365
1370
1375
1380
1385
1390
1395
1400
1405
1410
1415
1420
1425
1430
1435
1440
1445
1450
1455
1460
1465
1470
1475
1480
1485
1490
1495
1500
1505
1510
1515
1520
1525
1530
1535
1540
1545
1550
1555
1560
1565
1570
1575
1580
1585
1590
1595
1600
1605
1610
1615
1620
1625
1630
1635
1640
1645
1650
1655
1660
1665
1670
1675
1680
1685
1690
1695
1700
1705
1710
1715
1720
1725
1730
1735
1740
1745
1750
1755
1760
1765
1770
1775
1780
1785
1790
1795
1800
1805
1810
1815
1820
1825
1830
1835
1840
1845
1850
1855
1860
1865
1870
1875
1880
1885
1890
1895
1900
1905
1910
1915
1920
1925
1930
1935
1940
1945
1950
1955
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020
2025
2030
2035
2040
2045
2050
2055
2060
2065
2070
2075
2080
2085
2090
2095
2100
2105
2110
2115
2120
2125
2130
2135
2140
2145
2150
2155
2160
2165
2170
2175
2180
2185
2190
2195
2200
2205
2210
2215
2220
2225
2230
2235
2240
2245
2250
2255
2260
2265
2270
2275
2280
2285
2290
2295
2300
2305
2310
2315
2320
2325
2330
2335
2340
2345
2350
2355
2360
2365
2370
2375
2380
2385
2390
2395
2400
2405
2410
2415
2420
2425
2430
2435
2440
2445
2450
2455
2460
2465
2470
2475
2480
2485
2490
2495
2500
2505
2510
2515
2520
2525
2530
2535
2540
2545
2550
2555
2560
2565
2570
2575
2580
2585
2590
2595
2600
2605
2610
2615
2620
2625
2630
2635
2640
2645
2650
2655
2660
2665
2670
2675
2680
2685
2690
2695
2700
2705
2710
2715
2720
2725
2730
2735
2740
2745
2750
2755
2760
2765
2770
2775
2780
2785
2790
2795
2800
2805
2810
2815
2820
2825
2830
2835
2840
2845
2850
2855
2860
2865
2870
2875
2880
2885
2890
2895
2900
2905
2910
2915
2920
2925
2930
2935
2940
2945
2950
2955
2960
2965
2970
2975
2980
2985
2990
2995
3000
3005
3010
3015
3020
3025
3030
3035
3040
3045
3050
3055
3060
3065
3070
3075
3080
3085
3090
3095
3100
3105
3110
3115
3120
3125
3130
3135
3140
3145
3150
3155
3160
3165
3170
3175
3180
3185
3190
3195
3200
3205
3210
3215
3220
3225
3230
3235
3240
3245
3250
3255
3260
3265
3270
3275
3280
3285
3290
3295
3300
3305
3310
3315
3320
3325
3330
3335
3340
3345
3350
3355
3360
3365
3370
3375
3380
3385
3390
3395
3400
3405
3410
3415
3420
3425
3430
3435
3440
3445
3450
3455
3460
3465
3470
3475
3480
3485
3490
3495
3500
3505
3510
3515
3520
3525
3530
3535
3540
3545
3550
3555
3560
3565
3570
3575
3580
3585
3590
3595
3600
3605
3610
3615
3620
3625
3630
3635
3640
3645
3650
3655
3660
3665
3670
3675
3680
3685
3690
3695
3700
3705
3710
3715
3720
3725
3730
3735
3740
3745
3750
3755
3760
3765
3770
3775
3780
3785
3790
3795
3800
3805
3810
3815
3820
3825
3830
3835
3840
3845
3850
3855
3860
3865
3870
3875
3880
3885
3890
3895
3900
3905
3910
3915
3920
3925
3930
3935
3940
3945
3950
3955
3960
3965
3970
3975
3980
3985
3990
3995
4000
4005
4010
4015
4020
4025
4030
4035
4040
4045
4050
4055
4060
4065
4070
4075
4080
4085
4090
4095
4100
4105
4110
4115
4120
4125
4130
4135
4140
4145
4150
4155
4160
4165
4170
4175
4180
4185
4190
4195
4200
4205
4210
4215
4220
4225
4230
4235
4240
4245
4250
4255
4260
4265
4270
4275
4280
4285
4290
4295
4300
4305
4310
4315
4320
4325
4330
4335
4340
4345
4350
4355
4360
4365
4370
4375
4380
4385
4390
4395
4400
4405
4410
4415
4420
4425
4430
4435
4440
4445
4450
4455
4460
4465
4470
4475
4480
4485
4490
4495
4500
4505
4510
4515
4520
4525
4530
4535
4540
4545
4550
4555
4560
4565
4570
4575
4580
4585
4590
4595
4600
4605
4610
4615
4620
4625
4630
4635
4640
4645
4650
4655
4660
4665
4670
4675
4680
4685
4690
4695
4700
4705
4710
4715
4720
4725
4730
4735
4740
4745
4750
4755
4760
4765
4770
4775
4780
4785
4790
4795
4800
4805
4810
4815
4820
4825
4830
4835
4840
4845
4850
4855
4860
4865
4870
4875
4880
4885
4890
4895
4900
4905
4910
4915
4920
4925
4930
4935
4940
4945
4950
4955
4960
4965
4970
4975
4980
4985
4990
4995
5000
5005
5010
5015
5020
5025
5030
5035
5040
5045
5050
5055
5060
5065
5070
5075
5080
5085
5090
5095
5100
5105
5110
5115
5120
5125
5130
5135
5140
5145
5150
5155
5160
5165
5170
5175
5180
5185
5190
5195
5200
5205
5210
5215
5220
5225
5230
5235
5240
5245
5250
5255
5260
5265
5270
5275
5280
5285
5290
5295
5300
5305
5310
5315
5320
5325
5330
5335
5340
5345
5350
5355
5360
5365
5370
5375
5380
5385
5390
5395
5400
5405
5410
5415
5420
5425
5430
5435
5440
5445
5450
5455
5460
5465
5470
5475
5480
5485
5490
5495
5500
5505
5510
5515
5520
5525
5530
5535
5540
5545
5550
5555
5560
5565
5570
5575
5580
5585
5590
5595
5600
5605
5610
5615
5620
5625
5630
5635
5640
5645
5650
5655
5660
5665
5670
5675
5680
5685
5690
5695
5700
5705
5710
5715
5720
5725
5730
5735
5740
5745
5750
5755
5760
5765
5770
5775
5780
5785
5790
5795
5800
5805
5810
5815
5820
5825
5830
5835
5840
5845
5850
5855
5860
5865
5870
5875
5880
5885
5890
5895
5900
5905
5910
5915
5920
5925
5930
5935
5940
5945
5950
5955
5960
5965
5970
5975
5980
5985
5990
5995
6000
6005
6010
6015
6020
6025
6030
6035
6040
6045
6050
6055
6060
6065
6070
6075
6080
6085
6090
6095
6100
6105
6110
6115
6120
6125
6130
6135
6140
6145
6150
6155
6160
6165
6170
6175
6180
6185
6190
6195
6200
6205
6210
6215
6220
6225
6230
6235
6240
6245
6250
6255
6260
6265
6270
6275
6280
6285
6290
6295
6300
6305
6310
6315
6320
6325
6330
6335
6340
6345
6350
6355
6360
6365
6370
6375
6380
6385
6390
6395
6400
6405
6410
6415
6420
6425
6430
6435
6440
6445
6450
6455
6460
6465
6470
6475
6480
6485
6490
6495
6500
6505
6510
6515
6520
6525
6530
6535
6540
6545
6550
6555
6560
6565
6570
6575
6580
6585
6590
6595
6600
6605
6610
6615
6620
6625
6630
6635
6640
6645
6650
6655
6660
6665
6670
6675
6680
6685
6690
6695
6700
6705
6710
6715
6720
6725
6730
6735
6740
6745
6750
6755
6760
6765
6770
6775
6780
6785
6790
6795
6800
6805
6810
6815
6820
6825
6830
6835
6840
6845
6850
6855
6860
6865
6870
6875
6880
6885
6890
6895
6900
6905
6910
6915
6920
6925
6930
6935
6940
6945
6950
6955
6960
6965
6970
6975
6980
6985
6990
6995
7000
7005
7010
7015
7020
7025
7030
7035
7040
7045
7050
7055
7060
7065
7070
7075
7080
7085
7090
7095
7100
7105
7110
7115
7120
7125
7130
7135
7140
7145
7150
7155
7160
7165
7170
7175
7180
7185
7190
7195
7200
7205
7210
7215
7220
7225
7230
7235
7240
7245
7250
7255
7260
7265
7270
7275
7280
7285
7290
7295
7300
7305
7310
7315
7320
7325
7330
7335
7340
7345
7350
7355
7360
7365
7370
7375
7380
7385
7390
7395
7400
7405
7410
7415
7420
7425
7430
7435
7440
7445
7450
7455
7460
7465
7470
7475
7480
7485
7490
7495
7500
7505
7510
7515
7520
7525
7530
7535
7540
7545
7550
7555
7560
7565
7570
7575
7580
7585
7590
7595
7600
7605
7610
7615
7620
7625
7630
7635
7640
7645
7650
7655
7660
7665
7670
7675
7680
7685
7690
7695
7700
7705
7710
7715
7720
7725
7730
7735
7740
7745
7750
7755
7760
7765
7770
7775
7780
7785
7790
7795
7800
7805
7810
7815
7820
7825
7830
7835
7840
7845
7850
7855
7860
7865
7870
7875
7880
7885
7890
7895
7900
7905
7910
7915
7920
7925
7930
7935
7940
7945
7950
7955
7960
7965
7970
7975
7980
7985
7990
7995
8000
8005
8010
8015
8020
8025
8030
8035
8040
8045
8050
8055
8060
8065
8070
8075
8080
8085
8090
8095
8100
8105
8110
8115
8120
8125
8130
8135
8140
8145
8150
8155
8160
8165
8170
8175
8180
8185
8190
8195
8200
8205
8210
8215
8220
8225
8230
8235
8240
8245
8250
8255
8260
8265
8270
8275
8280
8285
8290
8295
8300
8305
8310
8315
8320
8325
8330
8335
8340
8345
8350
8355
8360
8365
8370
8375
8380
8385
8390
8395
8400
8405
8410
8415
8420
8425
8430
8435
8440
8445
8450
8455
8460
8465
8470
8475
8480
8485
8490
8495
8500
8505
8510
8515
8520
8525
8530
8535
8540
8545
8550
8555
8560
8565
8570
8575
8580
8585
8590
8595
8600
8605
8610
8615
8620
8625
8630
8635
8640
8645
8650
8655
8660
8665
8670
8675
8680
8685
8690
8695
8700
8705
8710
8715
8720
8725
8730
8735
8740
8745
8750
8755
8760
8765
8770
8775
8780
8785
8790
8795
8800
8805
8810
8815
8820
8825
8830
8835
8840
8845
8850
8855
8860
8865
8870
8875
8880
8885
8890
8895
8900
8905
8910
8915
8920
8925
8930
8935
8940
8945
8950
8955
8960
8965
8970
8975
8980
8985
8990
8995
9000
9005
9010
9015
9020
9025
9030
9035
9040
9045
9050
9055
9060
9065
9070
9075
9080
9085
9090
9095
9100
9105
9110
9115
9120
9125
9130
9135
9140
9145
9150
9155
9160
9165
9170
9175
9180
9185
9190
9195
9200
9205
9210
9215
9220
9225
9230
9235
9240
9245
9250
9255
9260
9265
9270
9275
9280
9285
9290
9295
9300
9305
9310
9315
9320
9325
9330
9335
9340
9345
9350
9355
9360
9365
9370
9375
9380
9385
9390
9395
9400
9405
9410
9415
9420
9425
9430
9435
9440
9445
9450
9455
9460
9465
9470
9475
9480
9485
9490
9495
9500
9505
9510
9515
9520
9525
9530
9535
9540
9545
9550
9555
9560
9565
9570
9575
9580
9585
9590
9595
9600
9605
9610
9615
9620
9625
9630
9635
9640
9645
9650
9655
9660
9665
9670
9675
9680
9685
9690
9695
9700
9705
9710
9715
9720
9725
9730
9735
9740
9745
9750
9755
9760
9765
9770
9775
9780
9785
9790
9795
9800
9805
9810
9815
9820
9825
9830
9835
9840
9845
9850
9855
9860
9865
9870
9875
9880
9885
9890
9895
9900
9905
9910
9915
9920
9925
9930
9935
9940
9945
9950
9955
9960
9965
9970
9975
9980
9985
9990
9995
9999
10000
10005
10010
10015
10020
10025
10030
10035
10040
10045
10050
10055
10060
10065
10070
10075
10080
10085
10090
10095
10100
10105
10110
10115
10120
10125
10130
10135
10140
10145
10150
10155
10160
10165
10170
10175
10180
10185
10190
10195
10200
10205
10210
10215
10220
10225
10230
10235
10240
10245
10250
10255
10260
10265
10270
10275
10280
10285
10290
10295
10300
10305
10310
10315
10320
10325
10330
10335
10340
10345
10350
10355
10360
10365
10370
10375
10380
10385
10390
10395
10400
10405
10410
10415
10420
10425
10430
10435
10440
10445
10450
10455
10460
10465
10470
10475
10480
10485
10490
10495
10500
10505
10510
10515
10520
10525
10530
10535
10540
10545
10550
10555
10560
10565
10570
10575
10580
10585
10590
10595
10600
10605
10610
10615
10620
10625
10630
10635
10640
10645
10650
10655
10660
10665
10670
10675
10680
10685
10690
10695
10700
10705
10710
10715
10720
10725
10730
10735
10740
10745
10750
10755
10760
10765
10770
10775
10780
10785
10790
10795
10800
10805
10810
10815
10820
10825
10830
10835
10840
10845
10850
10855
10860
10865
10870
10875
10880
10885
10890
10895
10900
10905
10910
10915
10920
10925
10930
10935
10940
10945
10950
10955
10960
10965
10970
10975
10980
10985
10990
10995
11000
11005
11010
11015
11020
11025
11030
11035
11040
11045
11050
11055
11060
11065
11070
11075
11080
11085
11090
11095
11100
11105
11110
11115
11120
11125
11130
11135
11140
11145
11150
11155
11160
11165
11170
11175
11180
11185
11190
11195
1120
```

The RR-Tree is mainly for propagating a datum to all tree nodes which will perform different operations to this datum depending on their positions in the tree. For computer graphics, this pattern is useful because it allows each PE to generate  $A^T X$  simultaneously where A is a constant and X is pid0. Evaluating  $A^T X$  in parallel allows the fast generation of a line; this will be further discussed in another pattern called Diagonal-Span-Tree (P12).

In general, a reverse tree is used to convert a symbolic representation in a parameter space to an iconic representation in image space, then the algorithm is performed iconically in a massive parallelism available in the polymorphic mesh.

Although the control algorithm uses the PE with the highest pid0 as the root of the tree, the PE with the lowest pid0 can be used as the root as well and the control algorithm is of equivalent complexity.

*(P6) Reverse Column Trees (RC-Tree)*

Similar to RR-Tree, the RC-tree can be formed by using pid1 as the control and N-S as the path to establish the tree connection. This is shown in the following control algorithm.

20

```

RC-TREE () {
    int t; /* t is time step */
    int pid1; /* row position of a PE */
    int M, logM; /* M is the side size of the mesh and logM=log M */
    int treemask=M%2; /* a flag to construct the tree */
    int mask; /* an intermediate condition */

    for (t=0; t< logM; t++) {
        mask = ANDALLBIT (~treemask | pid1 );
        if(~mask)
            {SHORT_NS; DISABLE;}
        if(mask && pid1<logM-t-1>
            N = MEM;
        if(mask && ~pid1<logM-t-1>

```

55

```

      MEM = S;

5      treemask = ASHIFT ( treemask, 1);

      }

10     }

```

The properties of the RC-Trees are the same RR-Trees except that RC-Trees are related to the data in the column of the mesh.

15 (P7) Row-Bus

For broadcasting purpose, a bus is a very useful pattern whose broadcasting distance is the shortest. One bus can be formed for every row of the polymorphic mesh by the following control algorithm.

```

20     ROW_BUS ()

25     {
        int sender; /* ID for the sender */

        int pid0;

30     SHORT_WE;

35     if (pid0 == sender)

40     E = MEM;

45     else
        MEM = W;
    }

```

A PE in a row is designated as the "sender" and the rest of the PEs are the receivers. All PEs "SHORT" their E-W path to establish the bus, and the sender will send the data to E (or W) while the receivers can receive the data from W (or E). (In another case when a datum is injected into W of E by the external controller, there is no "sender" PE and all PEs are the "receivers".)

Using the mechanism provided by the connection control block, "sender" is loaded to SRM. The "INVERTed" SRM is "XORed" with register X which stores pid0; the resulting bits are "ANDALLBITed". A '1' in XANDALL identifies the PE as a sender and those PEs with XANDALL = 0 are the receivers.

*(P8) Column Bus*

Similar to Row Bus, a Column Bus can be formed for each column of the polymorphic mesh by using pid1 as the control and N-S as the path as shown in the following control algorithm.

5

**COLUMN\_BUS 0**

{

10

int sender; /\* ID for the sender \*/

int pid1;

15

**SHORT\_NS;**

20

if (pid1 == sender)

S = MEM;

25

else

MEM = N;

}

30

The property of the column bus is the same as the row bus.

In combination, the row bus and the column bus can be used to broadcast a common datum to all PEs in the mesh in two steps. At the first step, the common datum can be broadcasted to all PEs in the top row; then at the second step, the PE at the top row can broadcast the common datum to all other PEs along column direction.

*(P9) Pyramid*

40

Pyramid configuration is powerful in image processing and computer vision mainly because of its capability in handling multi-resolution images. This pattern can be formed from the mesh by the following control algorithm:

45

50

55

5

```

PYRAMID ()

10 {
    int t; /* t is time step */

15     int pid0; /* column position of a PE */

20     int pid1; /* row position */

     int M, logM; /*M is the side size of the mesh and logM=log M*/
25     int hmask=1, vmask=1; /*two flags to construct the pyramid*/

30     for (t=0; t< logM; t++) {
        /*cycle 1 action*/
        if(~hmask | ~vmask)
            {SHORT__WE; SHORT__NS; DISABLE;}
        if(hmask && vmask && ~pid0<t> && ~pid1<t>)
35         E = MEM;
        if(hmask && vmask && pid0<t> && ~pid1<t>)
40         {N = MEM; MEM1 = W;}
        if(hmask && vmask && ~pid0<t> && pid1<t>)
45         E = MEM;
        if(hmask && vmask && pid0<t> && pid1<t>)
50         {MEM0 = N; MEM2 = W;}
    }
}

```

55

```

/*cycle 2 action*/

5      if(~hmask | ~vmask)
          {SHORT_WE; SHORT_NS; DISABLE;}
          if(hmask && vmask && ~pid0<t> && ~pid1<t>)
10     NO_ACTION;
          if(hmask && vmask && pid0<t> && ~pid1<t>)
15     S = MEM1;
          if(hmask && vmask && ~pid0<t> && pid1<t>)
             NO_ACTION;
20     if(hmask && vmask && pid0<t> && pid1<t>)
             MEM1 = N;
25
            hmask = hmask & pid0<t>;
            vmask = vmask & pid1<t>;
30     }
}

```

35 The control algorithm consists of  $\log M$  steps and within each step there are two control cycles. In another word, each step forms a level of the pyramid in two PE cycles.

Figure 12, 13 and 14 depict the pyramid control algorithm by a sample 8x8 mesh. Two masks, hmask (for row) and vmask (for column), are initialized as '1' such that all PEs in the mesh are 'enabled' at the first time step. At  $t=0$ , all PEs are active and every 2x2 PEs are formed as a group. These four (2x2) PEs are the NW, NE, SW and SE sons of the pyramid and the parent is the same as the SE son. The activity of the four sons are distinguished by the  $pid0<0>$  and  $pid1<0>$  bits. The SE son (or the parent) being designated as  $pid0<0> = pid1<0> = 1$ , will receive data from the SW ( $pid0<0> = 0$  and  $pid1<0> = 1$ ) and NE ( $pid0<0> = 1$  and  $pid1<0> = 0$ ) sons at the first cycle. In this cycle, the NW son enroutes its data to the NE son; this data will be received by the parent at the second cycle. At the second cycle, only the NE son and the parent are involved in sending and receiving; the other two PEs have no action. Both vmask and hmask are updated to control the connection of next time step.

At  $t=1$ , again four PEs form the four sons and one parent of the next-level pyramid. But these four PEs span in a 4x4 mesh as shown in Figure 13. The activity of four sons and the parent is the same as  $t=0$  except that PEs at even rows or even columns are disabled. These disabled "non-pyramid" PEs SHORT their W to E line, and N to S line to establish the pyramid connection.

PEs that constitute the last-level pyramid are shown in Figure 14. Their activities are the same as the previous two steps.

Orthogonal to the pyramidal structure described above, the pyramid pattern has a mesh connection at each level. That is for each node in the pyramid there exist four neighbors (N, S, E, W) at the same level other than four sons at the level below and one parent at the level above. This relation is shown in Figure 15.

The control algorithm for obtaining the neighbors at the same level has been imbedded in the above pyramid control algorithm. For example, at t=0 as shown in Figure 8a, the neighbors at the same level are connected by the original mesh. At t=1, the neighbors at the same level are scattered in every other row and column and the mesh connection for them has been established by the above-mentioned control algorithm PYRAMID.

To obtain the content of its neighbors at the same level of the pyramid, two control cycles are added to every step of the control algorithm PYRAMID as follows. Cycle 3 is to obtain content of N and W while cycle 4 is to obtain that of S and E.

10

15

20

**PYRAMID ()**

25

{

int t; /\* t is time step\*/

30

int pid0; /\* column position of a PE\*/

int pid1; /\* row position\*/

35

int M, logM; /\*M is the side size of the mesh and logM=log M\*/

int hmask=1, vmask=1; /\*two flags to construct the pyramid\*/

40

for (t=0; t&lt; logM; t++) {

/\*cycle 1 action\*/

45

if(~hmask | ~vmask)

{SHORT\_\_WE; SHORT\_\_NS; DISABLE;}

if(hmask &amp;&amp; vmask &amp;&amp; ~pid0&lt;t&gt; &amp;&amp; ~pid1&lt;t&gt;)

50

55

5

```
E = MEM;
```

10

```
if(hmask && vmask && pid0<t> && ~pid1<t>)
```

```
{N = MEM; MEM1 = W;}
```

15

```
if(hmask && vmask && ~pid0<t> && pid1<t>)
```

```
E = MEM;
```

20

```
if(hmask && vmask && pid0<t> && pid1<t>)
```

```
{MEM0 = N; MEM2 = W;}
```

25

/\*cycle 2 action\*/

30

```
if(~hmask | ~vmask)
```

```
{SHORT_WE; SHORT_NS; DISABLE;}
```

35

```
if(hmask && vmask && ~pid0<t> && ~pid1<t>)
```

```
NO_ACTION;
```

40

```
if(hmask && vmask && pid0<t> && ~pid1<t>)
```

```
S = MEM1;
```

45

```
if(hmask && vmask && ~pid0<t> && pid1<t>)
```

```
NO_ACTION;
```

50

```
if(hmask && vmask && pid0<t> && pid1<t>)
```

```
MEM1 = N;
```

55

/\*cycle 3 action\*/

```
if(~hmask | ~vmask)
```

```

{SHORT__WE; SHORT__NS; DISABLE;}

5      if(hmask && vmask){

          S = MEM3;

          E = MEM4;

10     MEM3 = N;

          MEM4 = W;

15     }

20     /*cycle 4 action*/
21
22     if(~hmask | ~vmask)

          {SHORT__WE; SHORT__NS; DISABLE;}

25     if(hmask && vmask){

          N = MEM5;

          W = MEM6;

          MEM5 = S;

          MEM6 = E;

35     }

40     hmask = hmask & pid0<t>;
41     vmask = vmask & pid1<t>;
42
43     }
44
45   }

```

50 Using the mechanism in the connection control block, hmask and vmask are loaded to F1 and F2 respectively. The pid0 in register X and pid1 in register Y are copied to SRX and SRY respectively. Shifting logically to the right, BSRX and BSRY will contain pid0<t> and pid1<t> at time step t. These two condition bits along with F1 and F2 are used to implement the SHORT actions.

55 The above-described pyramid has a base of MxM and a shrinkage of 2, meaning that the level above the base has  $M/2 \times M/2$  PEs and so on. The PYRAMID control algorithm can be extended to handle any shrinkage K, where K is a power of 2, by updating hmask and vmask by  
 $hmask = hmask \& pid0<t> \& pid0<t+1>$   
 $vmask = vmask \& pid1<t> \& pid1<t+1>$   
and by skipping the pyramid node actions on every odd t step.

## (P10) Reverse Pyramid

Information flows from bottom to top level in a pyramid for iconic to symbolic conversion. However, there is need for the information to flow in the opposite direction for symbolic to iconic conversion. This can be served by the Reverse Pyramid (R-Pyramid) formed from the polymorphic mesh by the following control algorithm.

```

R-PYRAMID ()
10
{
    int t; /* t is time step*/
15
    int pid0; /* row position of a PE*/
    int pid1; /* column position of a PE*/
20
    int M, logM; /*M is the side size of the mesh and logM=log M*/
    int mask=M%2; /* a flag to construct the pyramid*/
    int hmask, vmask;
25

    for (t=0; t< logM; t++) {
30
        hmask = ANDALLBIT (~mask | pid0 );
        vmask = ANDALLBIT (~mask | pid1 );
35

        /*cycle 1*/
40
        if(~hmask | ~vmask)
            {SHORT__WE; SHORT__NS; DISABLE;}
        if(hmask && vmask && pid0<logM-t-1> && pid1<logM-t-1>)
45
            N = MEM2;
        if(hmask && vmask && pid0<logM-t-1> && ~pid1<logM-t-1>)
            MEM2 = S;
50
        if(hmask && vmask && ~pid0<logM-t-1> && pid1<logM-t-1>)
            NO__ACTION;
55
        if(hmask && vmask && ~pid0<logM-t-1> && ~pid1<logM-t-1>)
}

```

NO-ACTION;

5

```

/*cycle 2*/
if(~hmask | ~vmask)
10 {SHORT_WE; SHORT_NS; DISABLE;}
if(hmask && vmask && pid0<logM-t-1> && pid1<logM-t-1>)
15 {N = MEM1; W = MEM3;}
if(hmask && vmask && pid0<logM-t-1> && ~pid1<logM-t-1>)
20 {MEM1 = S; W = MEM2;}
if(hmask && vmask && ~pid0<logM-t-1> && pid1<logM-t-1>)
25 MEM3 = E;
if(hmask && vmask && ~pid0<logM-t-1> && ~pid1<logM-t-1>)
30 MEM2 = E;
mask = ASHIFT ( mask, 1);
35 }
}

```

The control algorithm for the R\_PYRAMID is a reverse process of the PYRAMID control algorithm and  
40 is an expansion of the RR\_TREE and RC\_TREE.

One half of the mesh size (the "mask") is loaded to register SRM, while pid0 in X and pid1 in Y are  
45 copied to SRX and SRY respectively. The "INVERTed" SRM is "ORed" with X then "ANDALLBITed" to  
produce a flag "hmask" in XANDALL. Similarly, "vmask" is produced in YANDALL. Along with these two  
condition bits, pid0 and pid1 are shifted logically from left to right to produce pid0<logM-t-1> and  
pid1<logM-t-1> at time step t in BSRX and BSYR respectively. Figure 16, 17 and 18 illustrate the forming  
of a pyramid in an 8x8 polymorphic pyramid.

In a similar way as PYRAMID, each step of the R-PYRAMID control algorithm consists of two cycles:  
50 the first cycle is an intermediate stage of sending data to the NW son (data are routed to the NE son in the  
cycle and to NW son at the second cycle); and at the second cycle while the NE son is routing the data to  
the NW son, the parent sends data to the NE and SW sons at the same time.

Connection for the neighbors at the same level of the pyramid can also be established by the R-  
PYRAMID control algorithm similar to PYRAMID. However, communication at the same level is not used in  
general for information flowing top-down. When it is necessary, actions similar to cycle 3 and 4 of the  
PYRAMID control algorithm can be used.

55 The above-described pyramid has a base of MxM and a shrinkage of 2, meaning that the level above  
the base has M/2 x M/2 PEs and so on. The R-PYRAMID control algorithm can be extended to handle any  
shrinkage K, where K is a power of 2, by initializing SRM by M/(logK) and shifting SRM log K bits per step.

## (P11) Cube

5 Cube is the most natural extension of the polymorphic mesh into a 3D structure. The usefulness of cube to 3D data structure (e.g. 3D image element = volume = voxel) can be analogously described as the usefulness of the mesh to 2D data structure (e.g. 2D picture element = area = pixel).

10 To form the cube from the mesh, the data structure in the third dimension is sliced vertically and allocated to one PE such that the communication in the third dimension can be accomplished by the local memory communication while the communication involving the other two dimensions are served by the mesh. In fact, the polymorphic feature is not used in forming this pattern and the cube formation may not be as novel as the other eleven patterns. Nevertheless, the cube pattern is supported by the polymorphic mesh with the saving of the connection pins in the third dimension. Since the saving is significant ( $2 \times M \times M$  pins in total), the forming of the cube from the polymorphic mesh is important to its VLSI implementation.

15 With the data slicing, and  $M \times M \times K$  cube can be formed, where  $K$  is an integer and the value of  $K$  is only limited by the amount of local memory in the PE.

15

## (P12) Diagonal-Span Tree (DST)

20 A Diagonal-Span-Tree (DST) is a binary tree whose leafs span the diagonal of the mesh once and only once. By this definition, the DST in an  $N \times N$  mesh has  $N$  leafs each of which occupies a diagonal node designated as  $PE(k, N-1-k)$ ,  $k=0$  to  $N-1$ . There are many possible DSTs in a mesh. We choose the one shown in Figure 19 (exemplified by a 3-level DST in an  $8 \times 8$  mesh) because it is simple to control.

25 As shown in Figure 19, the root of the DST is at  $PE(0, 0)$  (upper left corner of the mesh). The left son of the root is four units away vertically (i.e.  $PE(4, 0)$ ) while its right son is four units away horizontally (i.e.  $PE(0, 4)$ ).

The second level sons are two units away from the corresponding first level sons vertically and horizontally. Thus  $PE(6, 0)$  and  $PE(2, 4)$  are the sons of  $PE(0, 4)$ , and  $PE(4, 2)$  and  $PE(6, 0)$  are the sons of  $PE(4, 0)$ .

30 Spanning in a similar way, all diagonal PEs of the mesh are the third level sons.

In a general definition, a Upper Left DST (ULDST) in a  $N \times N$  mesh has  $PE(0, 0)$  as the root and the diagonal PEs ( $k, N-1-k$ ),  $k=0$  to  $N-1$ , as the leafs. The  $i$ -th level ( $i=1$  to  $\log N$ ) left son of  $PE(s, t)$  is  $PE(s + 2^{i-1}(\log N - i), t)$  and the  $i$ -th level right son of  $PE(s, t)$  is  $PE(s, t + 2^{i-1}(\log N - i))$ .

35 The control algorithm for ULDST is listed below.

35

40

## ULDST ()

45 {

```
int fs=0, fr=0; /*flag-send is used to construct the DST*/
```

50

55

```

/*flag-receive is an intermediate var to update fs*/

5      int pid0, pid1;

int t; /* t is time step */

10     int M, logM; /*M is the side size of the mesh and logM=log M*/
      _int treemask=M%2; /* a flag to construct the tree*/

15     if(pid0==0 && pid1==0) {

20       fs = 1; fr = 1;

25       for (t=0; t< logM; t++) {

30         hmask = ANDALLBIT (~treemask | pid0 );
         vmask = ANDALLBIT (~treemask | pid1 );

35         if(~hmask | ~vmask)

40         {SHORT_WE; SHORT_NS; DISABLE; }

45         if(hmask | vmask && fr)
           {E = fs; S = fs; fr = 0; }

50         if(hmask | vmask && ~fr)
           {fs = W | N; fr = W | N; }

55         treemask = ASHIFT ( treemask, 1);

60       }

65     }

```

Using an 8x8 polymorphic mesh as an example, the ULDST algorithm can be explained as follows. A root PE(000, 000) is selected for the DST and its fs and fr are set to 1. At t=0 time step, rows 000 and 100, and columns 000 and 100 are active while the rest are disabled. The disabled PEs short their WE and NS to establish a temporary path for updating fs. The active PE with fr=1 sends its fs value to E and S neighbors, then reset fr to 0 so that it will not be a sender at the next time step. The receivers (with fr=0) updates its fs as the ORed value of N and W. The receivers also update its fr with the same ORed result therefore the PE just received a 1 from N or W will be the sender at the next time step. Step t=0 selects two PEs (PE(000, 100) and (100,000)) as nodes of DST (via setting their fs=1) furthermore this step prepares them as the new senders (via setting fr=1) to set more DST nodes in the following step.

At the next step, each of the two new senders will produce two DST nodes and two senders in a similar way. The new nodes and senders are PEs (000, 110), (010, 100), (100, 010) and (110, 000).

At  $t=2$ , the diagonal PEs are reached by the control algorithm; their  $fs$  will be set to 1 to identify themselves as part of the DST. Furthermore, their  $fr$  will be set to 1, which is additional information to identify themselves as diagonal nodes. The diagonal identification is a bonus from the DST algorithm. It is useful for many types of computing to be discussed in the next section.

- 5 To form an DST, the flag  $fs$  is used as the condition: PEs with  $fs=0$  short We and NS paths while PEs with  $fs=1$  send MEM to E and S, and receive data from W and N.

Using the mechanism in the connection control block, "treemask" is loaded to SRM,  $fs$  to F1 and  $fr$  to F2. The INVERTed SRM is ORed with pid0 in register X and pid1 in register Y respectively; the "ORed" results are "ANDALLBITed" to produce "hmask" in XANDALL and "vmask" in YANDALL. At the end of 10 each step, SRM is shifted arithmetically one bit to the right. The SHORT action is then based on F2, XANDALL and YANDALL.

By choosing a different root and using a similar control algorithm, different DST can be formed in the same polymorphic mesh. Figure 20 shows a DST with the root in the lower right corner; it is called URDST.

- In the following we show that the coexistence of ULDST and LRDST allows us to compute  $A^T X + B^T Y + C$  15 in parallel for each pixel  $(x, y)$  in an image. This capability has a very wide applications in computer graphics and computer vision.

## APPLICATIONS

- 20 Besides the well-known application of a plain mesh to image processing the following applications of polymorphic mesh are either faster in polymorphic mesh or not-implemented in the plain mesh. These applications are categorized into six types.

- 25 (1) *Divide-and-conquer computing*

This type of computing involves in dividing a set of  $N$  data into two groups at first according to their property. Then by applying the same property, each group is further divided into two subgroups. This 30 process is repeated until each group contains only one datum.

For mesh connection, this type of computing has the complexity of  $O(N)$  or higher. By transforming to trees and pyramids, the complexity of this type of computing is  $O(\log N)$  in the polymorphic mesh. The speedup for a data set of  $N = 1024$  is 100:1, a two orders of magnitude improvement.

Computations belonging to this type include sorting, find maximum, minimum, k-th largest and median.

- 35 All these algorithms are of complexity  $O(\log N)$ .

### (2) *Iconic-to-symbolic Conversion*

- 40 This type of computing is specific to computer vision and is often called intermediate level processing. Given an image, we are interested to know

- (a) how many pixels satisfy a specified property;
- (b) which pixels satisfy the specified property;
- (c) are SOME or NON or ALL pixels satisfy the specified property;

- 45 The property for the above can be

- (a) equal to a value;
- (b) greater than a value
- (c) smaller than a value
- (d) condition synthesized arithmetically and logically from above

- 50 All these algorithms can be computed in  $O(\log N)$  steps in polymorphic mesh by tree and pyramid patterns. More importantly and in significant contrast to the conventional fixed-pattern approach, only the answer (as against the whole intermediate image) is output. This significantly reduces the I/O rate; in the extreme case, only one bit (YES/NO) as against  $1024 \times 1024$  bits (the whole image) is output.

- Related to I/O, extra N-S path has been traditionally added to the mesh to support concurrent I/O and 55 processing. This mechanism and benefit are also valid for the polymorphic mesh, however, is irrelevant to the invention.

**(3) Statistic Measurement**

The polymorphic mesh is capable of computing the following statistics in  $O(\log N)$  steps. The statistics include.

- 5      (a) mean, variance, standard deviation;
- (b) area, perimeter and centroid
- (c) first moment, second moment and cross moment

Item (a) is general to a set of  $N$  data while item (b) and (c) are specific to an image.

10     The statistics are foundations of other algorithms. In computer vision, they are the basis for region analysis and pattern recognition.

**(4) Compute  $A^T x + B^T y + C$** 

15     To compute  $A^T x + B^T y + C$ , four patterns need to be formed by the polymorphic mesh : they are one Upper-Left-Diagonal-Span-Tree (ULDST), one Lower-Right-DST (LRDST), Row-Buses and Column-Buses. The ULDST and LRDST must coexist to compute  $A^T x + C$  and  $B^T y$  simultaneously while the Row-Buses and the Column-Buses are coexistent to do the summing (e.g.  $A^T x + C + B^T y$ ).

20     The algorithm is performed in a bit-serial manner. The extra two trees in the pixel-plane can be eliminated. Constant arguments A and B are broadcast to all PEs before the computing begins and are stored in array A and B with  $(\log M - 1)$  "0"s preempted at the beginning of the arrays. The storage of A and B is bit-reversed so that after the preempted "0"s are accessed, the least significant bit of A and B will be accessed first. The constant argument C is injected into the polymorphic mesh through W of the root of the ULDST one bit per time step starting from the least significant bit. Using the ULDST as the tree to compute 25      $A^T x + C$ , each PE has three variables (sum, carry and delay). At each time step, "sum" is passed Eastbound and "delay" is passed Southbound while each PE performs two operations (a) add N with array A, store the carry bit in "carry" and (b) store N in "delay". After  $\log M$  steps, the diagonal PEs of the mesh (or the leafs of ULDST) stores  $A^T x$  for the corresponding row. Similarly, the computing of  $B^T y$  can be done by LRDST with "0" injected from E of the root and with each PE passes "delay" Northbound and "sum" Westbound. 30     After  $\log M$  steps, the diagonal elements store  $B^T y$  for each corresponding column.

After obtaining  $A^T x + C$  in rows and  $B^T y$  in columns, the polymorphic mesh changes to Row-buses in WE path and Column-buses in NS path. Each PE then adds the value on Row-bus to value on Column-bus to produce  $A^T x + B^T y + C$  in bit serial fashion.

35     Since there is a conflict of resource in establishing the DSTs and the buses simultaneously, the result of  $A^T x + B^T y + C$  is delivered in bit serial at every other time step.

**(5) Fast Line Detection**

40     With the capability of computing  $A^T x + B^T y + C$  in every two time steps of the polymorphic mesh, we can have every pixel  $(X, Y)$  in an  $M \times M$  image to decide whether it is on a given line determined by A, B and C. Assume that all numbers are K bits long, the decision can be made in  $\log M + 2K$  time steps.

45     The capability of fast line detection is very useful in computer graphics and computer vision. For computer graphics, it is useful in displaying convex polygons, in creating shadow, in clipping, in drawing spheres, in computing adaptive histogram equalization, in texture mapping and anti-aliasing. For computer vision, such a capability is useful in computing Fast Hough Transform for detecting lines in a noisy image.

**(6) Converting Symbolic Information to Iconic Information**

50     With the massive parallel hardware available in the polymorphic mesh, it is advantageous to convert a symbolic processing (usually not done in mesh) into iconic processing so that the processing can be done in a massively parallel way. The Fast Hough Transform mentioned above is one such example. The "mask generating" to be described is another class applications in this category.

**(6.1) Band Mask Generation**

A "band mask" is confined within two parallel lines, one of which is determined by  $(A, B, C1)$  and the other by  $(A, B, C2)$ . To generate a "band mask", each PE computes  $A^*X + B^*Y + C1$  and  $A^*X + B^*Y + C1 + -(C2-C1)$  as described above. The computing produces  $S1$  (the sign of  $A^*X + B^*Y + C1$ ) and  $S2$  (the sign of  $A^*X + B^*Y + C2$ ), both of which are used in deciding whether pixel  $(X, Y)$  is inside the band.

The "band mask" provides a capability to a computer vision system in processing only the region of interest. The human vision adopts a different strategy in generating masks. The "strategy" is a symbolic information but its processing is actually done iconically as described.

10

**(6.2) Polygonal Mask Generation**

The "polygonal mask" is a generalization of the band mask. It consists of the union of  $P$  half planes, each of which is determined by a line specified by  $A^*X + B^*Y + C$ . Using the line detection capability, we can obtain signs  $S1, S2$  to  $SP$  for the corresponding lines. The Boolean combination of  $S1$  to  $Sp$  determines a pixel  $(X, Y)$  is inside the polygon.

**20 CONCLUSIONS**

The preferred embodiment of the invention carries out the following transforms under control of the connection control mechanism:

- The physical  $M \times M$  mesh connection to one row and one column linear arrays, each  $M \times M$  long.
- 25 The physical  $M \times M$  mesh connection to  $M$  row trees, each of which has  $M$  leaves.
- The physical  $M \times M$  mesh connection to  $M$  column trees, each of which has  $M$  leaves.
- The physical  $M \times M$  mesh connection to a  $M \times M$  orthogonal tree.
- The physical  $M \times M$  mesh connection to  $M$  reverse row trees, each of which has  $M$  leaves.
- The physical  $M \times M$  mesh connection to  $M$  reverse column trees, each of which has  $M$  leaves.
- 30 The physical  $M \times M$  mesh connection to  $M$  row buses, each of which has  $M$  PEs.
- The physical  $M \times M$  mesh connection to  $M$  column buses, each of which has  $M$  PEs.
- The physical  $M \times M$  mesh connection to a pyramid with  $M \times M$  base and a shrinkage  $K$  where  $K$  is the power of 2.
- The physical  $M \times M$  mesh connection to a reverse pyramid with  $M \times M$  base and a shrinkage  $K$  where  $K$  is the power of 2.
- 35 The physical  $M \times M$  mesh connection to a  $M \times M \times K$  cube, where  $K$  is an integer and is only limited by the local memory of PE.

The invention carries out the following transform under control of programming outside the connection control mechanism:

- 40 The physical  $M \times M$  mesh connection to a DST tree (whose root can be at any corner of the mesh). Up to two DSTs can coexist if their roots are at the opposite corners of a diagonal.
  - The physical  $M \times M$  mesh connection to a  $M \times M$  orthogonal tree in  $O(M^{*}2)$  silicon area. A saving at a factor of  $O(\log M)^{*}2$  is obtained by the invention, where  $M$  is the side size of the orthogonal tree.
  - 45 The class of divide-and-conquer algorithms of linear,  $O(M)$ , complexity into logarithm,  $O(\log M)$ , complexity.
  - A saving of  $M/\log M$  is obtained by the invention. Such class of algorithms is discussed in the description of the preferred embodiment.
  - The iconic-to-symbolic conversion (intermediate level processing) occurs within the polymorphic mesh such that the I/O is significantly reduced. In an extreme case as discussed in the description of the preferred embodiment of the invention, a reduction of six orders of magnitude is obtained.
  - 50 Symbolic representation can be transformed into iconic representation by the above patterns such that the processing can be performed iconically in massive parallelism available in mesh. Such a feature expands the capability of mesh into the domain of symbolic processing.
- The mesh system is capable of:
- (a) performing iconic processing,
  - 55 (b) converting iconic information to symbolic information,
  - (c) converting symbolic information to iconic information and
  - (d) performing symbolic processing in its iconic equivalence.

The invention allows the computing of  $A'x + B'y + C$  in an  $M \times M$  mesh in  $O(\log M)$  steps for every pixel  $(x, y)$  in parallel where A, B and C are constant integers.

The invention allows the detection for every pixel  $(x, y)$  in an  $M \times M$  image whether the pixel  $(x, y)$  is (a) on the line (b) to the right of a line or (c) to the left of a line in  $O(\log M)$  step.

5 The invention allows the parallel detection for every pixel  $(x, y)$  of an  $M \times M$  image whether the pixel is inside or outside of a band where the band is formed by two parallel lines.

The invention allows the parallel detection for every pixel  $(x, y)$  of an  $M \times M$  image whether the pixel is inside or outside of a polygon.

10 The invention can be generalized for 3D mesh (physical cube) to form the higher-dimension-extension of the twelve patterns by adding a register Z, a shift register SRZ and a flag F3 to the connection control mechanism. (e.g. The higher-dimension extension of a cube is a 4D hypercube.)

The 3D image element is the voxel. The voxel is the volume element, analogous to the 2D pixel, which has area only. The 3D extension of the invention allows the parallel detection for each voxel  $(x, y, z)$  whether the voxel is inside or outside of

15 (a) a region formed by two parallel planes,

(b) a polyhedron, or

(c) whether the voxel is on, to the left or to the right of a plane.

The invention applies the concept of "polymorphic" to a physical mesh via a "connection control mechanism." The same concept and mechanism can be generalized to other physically-fixed-connections.

20 The pattern formed by the polymorphic mesh can be adaptive to the nature of the data by loading the F1 and/or F2 registers via the output of ALU.

Arbitrary patterns can be formed by the polymorphic mesh by setting F1 and F2 registers via instruction or memory. The instruction, the memory value, intermediate processing values from a neighboring processing element, and diagnostic information within the processing element are representative system

25 operation parameters which may be used to set the flag registers, by well known techniques and simple means not shown. The connection control mechanism thus comprises flag register means, which flag register means is settable as a function of system operation parameters, to provide control information usable in setting a new pattern into at least one of the pattern registers. This capability, accessible to the programmer, permits the programmer to set up adaptation to data-related and condition-related future event

30 possibilities. Upon occurrence of such an event, a flag register is set, and a new pattern fetched or calculated in response.

Thus, while the invention has been described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changes in form and details may be made without departing from the scope of the invention.

35

## Claims

1. An optimizable reconfigurable array processing system for performing under programmable control a series of programmably defined tasks upon input images, having differing optimal system configurations as a composite function of task definition and image input, so that at differing times under differing conditions there are identifiable differing optimal configurations, comprising:

40 system control means (3) including operator input means, image input means and operation control means, an array of polymorphic mesh processing elements (2), each comprising:

45 a memory (7);

an ALU (6) connected to said memory; and

connection control mechanism (8), with a finite number of simple connection paths to said ALU (6) and to a related subset of said array of polymorphic mesh processing elements (2), with means (10) to form selective internal and external interconnections of the polymorphic mesh processing element according to a

50 connection control pattern, and means (21,22,23) to provide a connection control pattern.

2. An optimizable reconfigurable array processing system according to Claim 1, wherein said connection control mechanism comprises pattern presentation means (21,22) making available to the processing element a surplus of pattern bit values defining a standard set of pattern values and an alternate set of pattern values, and pattern value selection means (23) for selecting said standard set of pattern values or,

55 alternatively, for selecting said alternate set of pattern values,

and said connection control mechanism also comprises switching means (10) responsive to the selected pattern bit values.

3. An optimizable reconfigurable array processing system according to Claim 2,  
 wherein said connection control mechanism (8) comprises a plurality of pattern registers (21,22), including a standard pattern register and an alternate pattern register, and a crossbar switch (10), with external connections to neighboring processing elements, with internal connections to said plurality of pattern  
 5 registers and to said ALU (6), and with control connections to said plurality of pattern registers; and  
 wherein said pattern presentation means comprises pattern register selection means (23), for selecting one pattern register;  
 and said means controlling said connection control mechanism in accordance with an optimization pattern includes gate switching means (16-20) responsive to the setting in the selected pattern register.
- 10 4. An optimizable reconfigurable array processing system according to Claim 3, wherein said connection control mechanism comprises two pattern registers (21,22), and said pattern register selection means for selecting one pattern register is a binary device (23).
- 15 5. An optimizable reconfigurable array processing system according to Claim 3, wherein said connection control mechanism comprises flag register means (31,32), said flag register means being settable as a function of system operation parameters, to provide control information usable in setting a new pattern into at least one of said pattern registers.
- 20 6. A dynamically optimizable reconfigurable array processing system comprising system control means including operator input means, image input means, operation control means and operation monitoring means controlling overall system operation, simultaneously monitoring so as to determine optimal system configuration as a composite function of operator input means and operation monitoring means, providing a signal defining an optimal configuration selection; and  
 an array of polymorphic mesh processing elements, each having a memory and an ALU, and each having a finite number of connections paths to related polymorphic mesh processing elements,  
 each having a polymorphic mesh connection control block having capability to form selective interconnection  
 25 of the polymorphic mesh processing element by short-circuiting selected connection paths, and by selecting a logical connective, and  
 means connecting said optimizing reconfiguration control signal to said polymorphic mesh connection control block.
- 30 7. A processing element for a dynamically optimizable reconfigurable array processing system comprising a multiplicity of processing elements generally controlled by a host computer to carry out image processing as a network, comprising:  
 memory means;  
 processing means;  
 I/O connection means;  
 35 internal connection means;  
 connection control means controlling the relationships of said other means in accordance with a control pattern; and  
 means to alter the control pattern in said connection control means.
- 40 8. A processing element for a dynamically optimizable reconfigurable array processing system according to Claim 7, wherein said means to alter the control pattern in said connection control means is means responsive to intermediate level processing in a related processing element.
9. A processing element for a dynamically optimizable reconfigurable array processing system according to Claim 7, wherein said means to alter the control pattern in said connection control means is means responsive to fault indication in a related processing element.
- 45 10. A processing element for a dynamically optimizable reconfigurable array processing system according to Claim 7, comprising in addition external memory data connection means (EMD 102) connecting to said memory means, processor means and connection control means (8).
- 50 11. A processing element for a dynamically optimizable reconfigurable array processing system according to Claim 10, comprising in addition a multiplexer and a plurality of bus connections for said external memory data connection to said memory means, said processing means and said connection control mechanism,  
 whereby said processing element may be operating local memory and external memory simultaneously.

FIG. 1

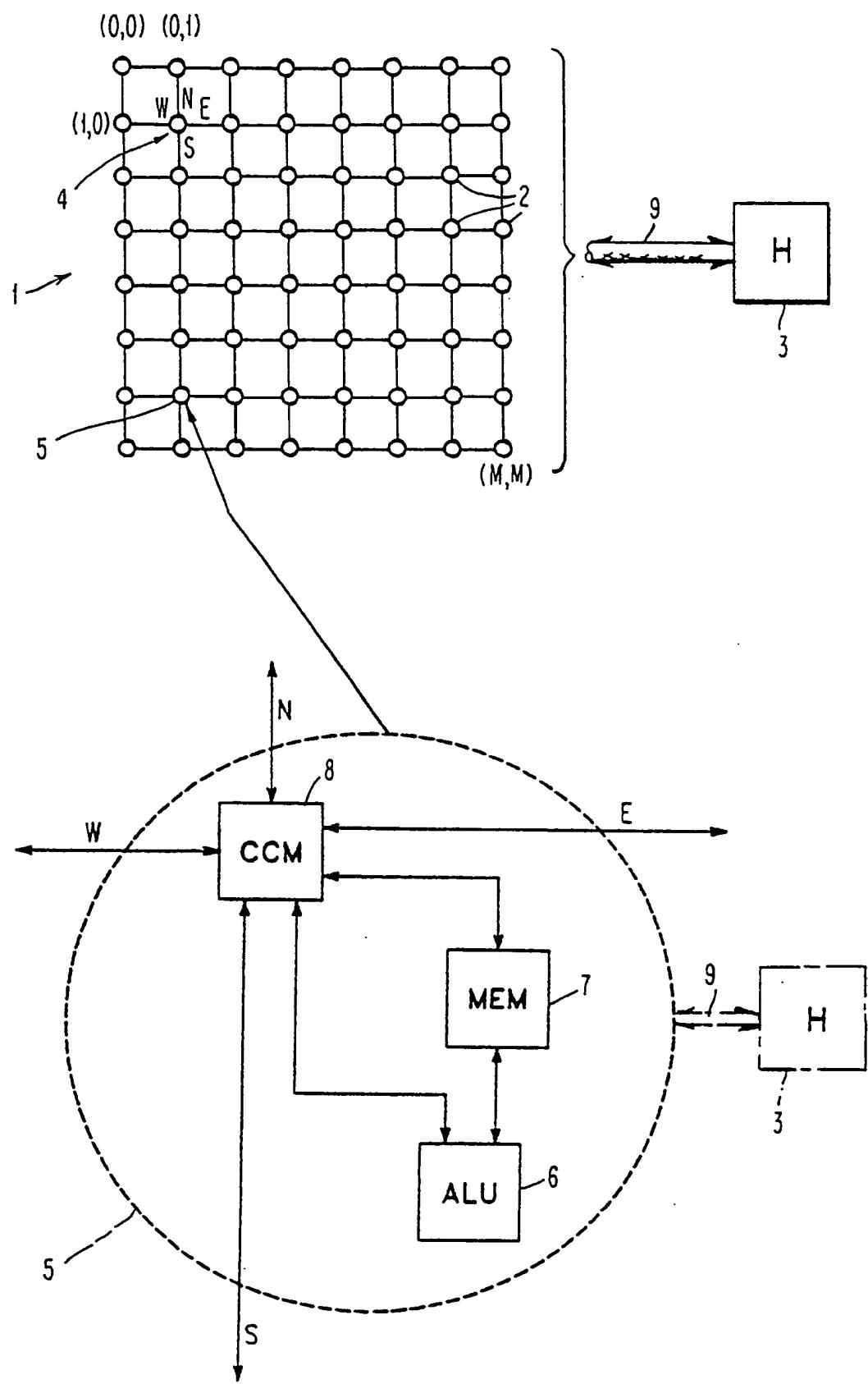


FIG. 2

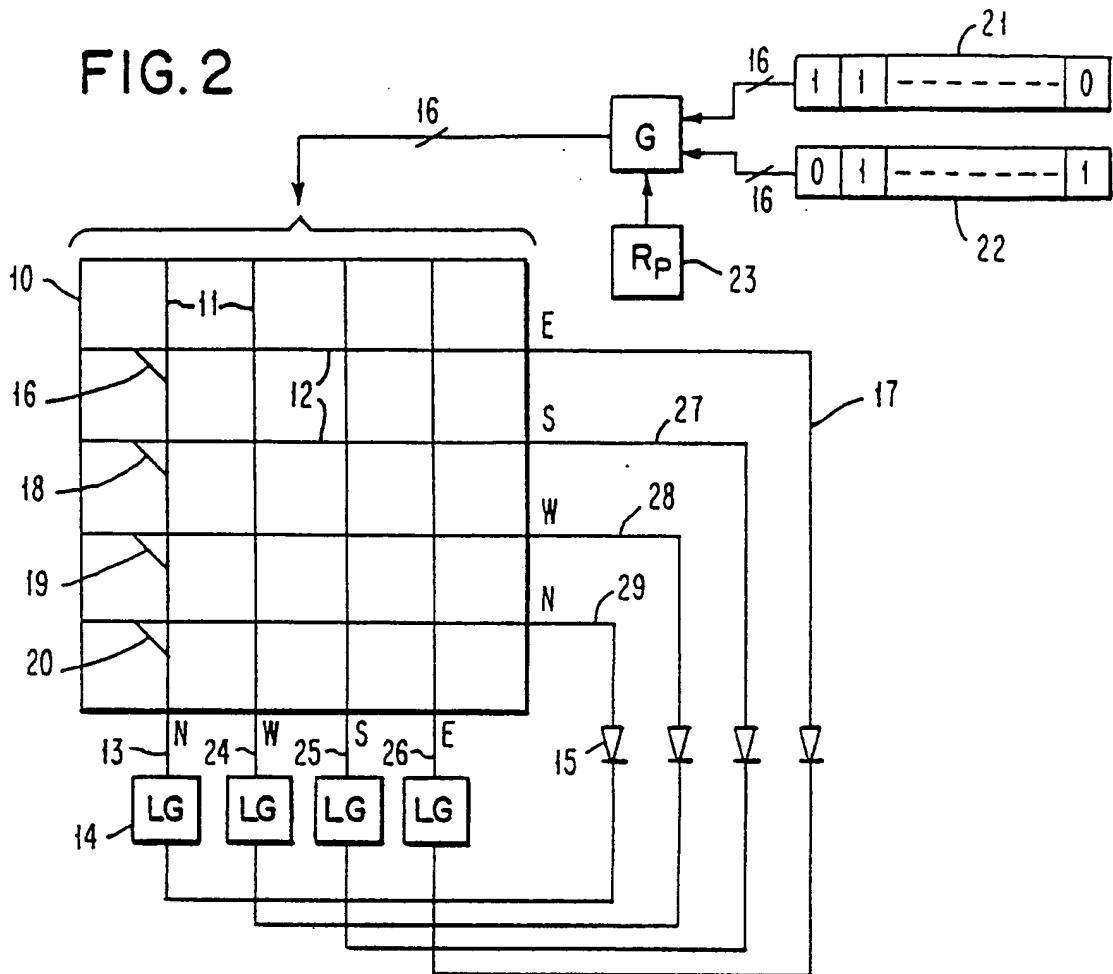


FIG. 3

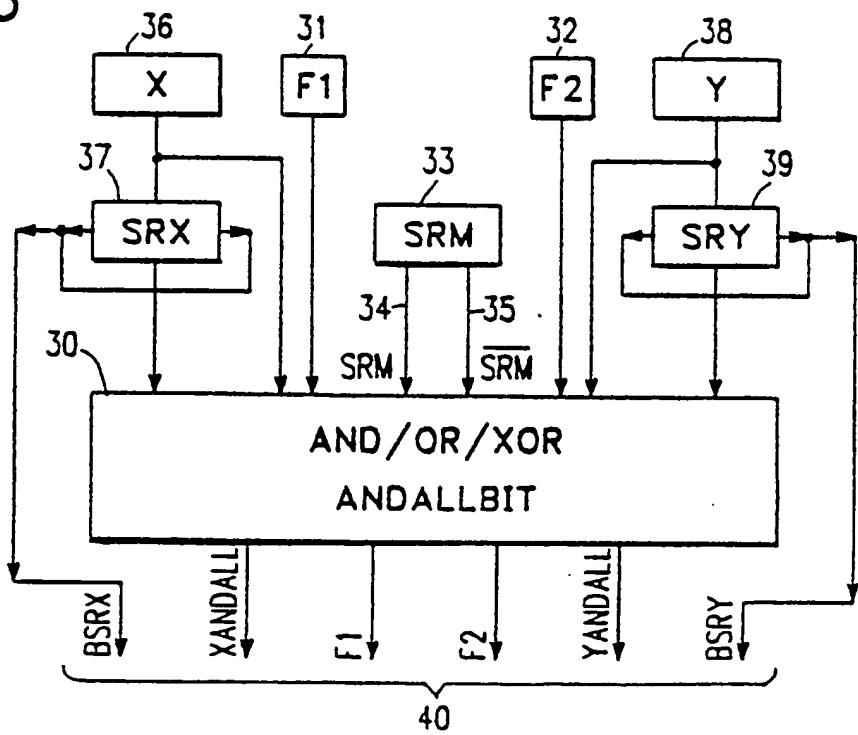


FIG. 4

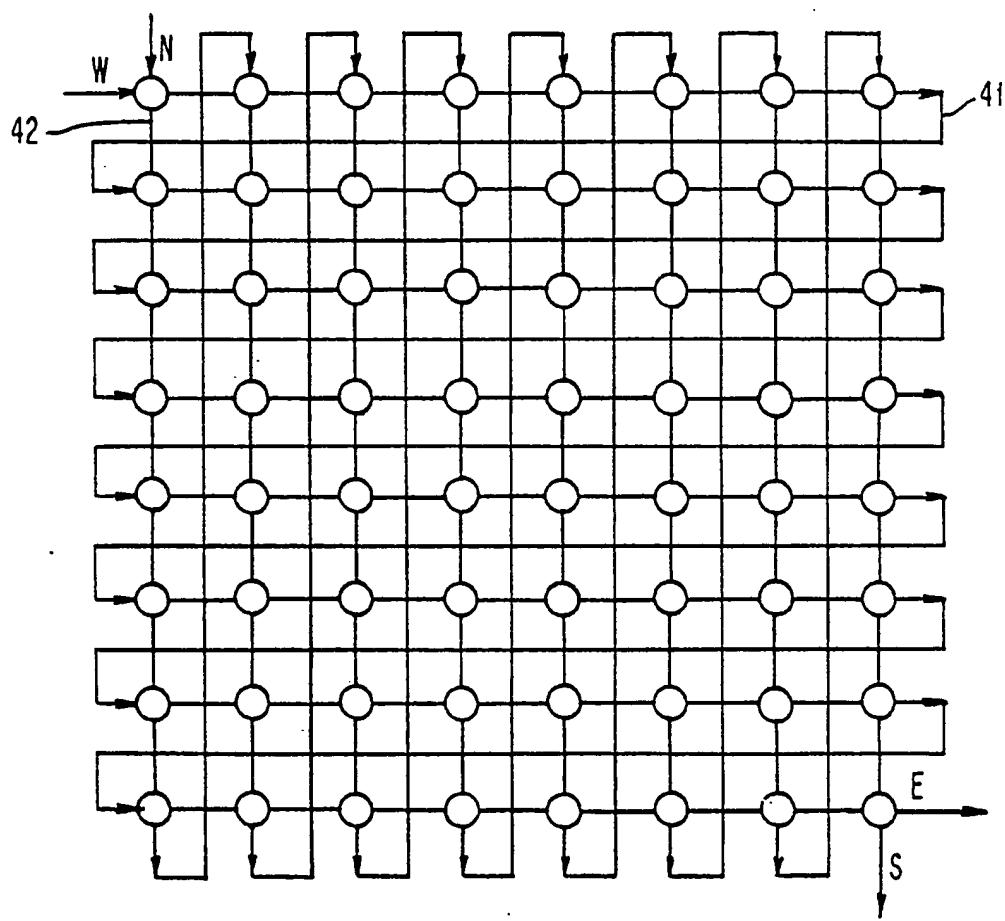


FIG. 5

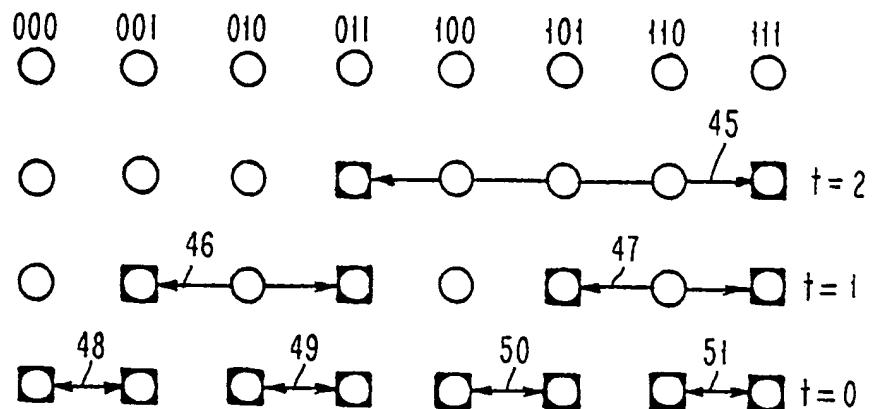


FIG. 6

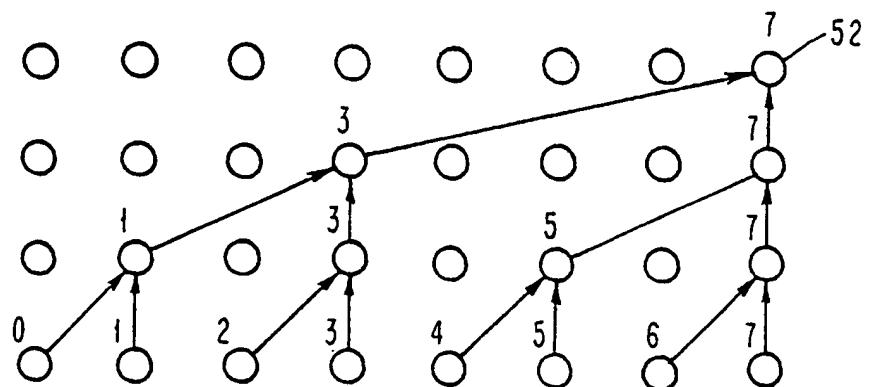


FIG. 7

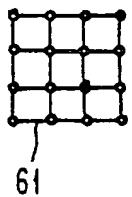


FIG. 9

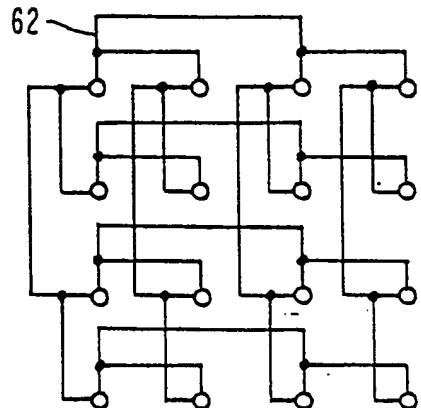


FIG. 8

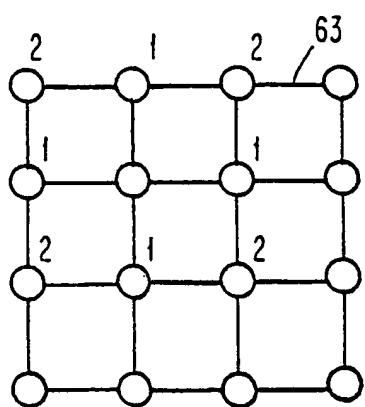


FIG. 10

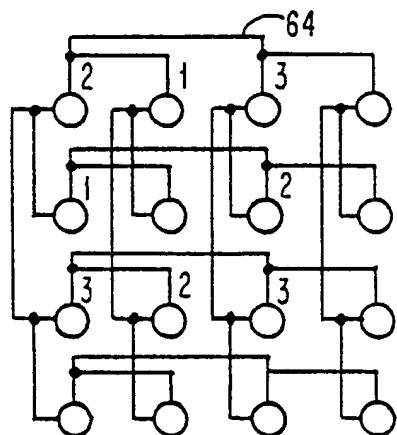


FIG. 11

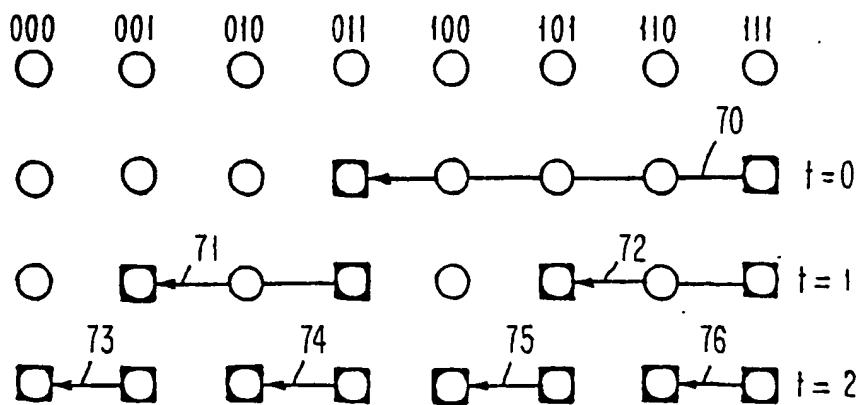


FIG. 12

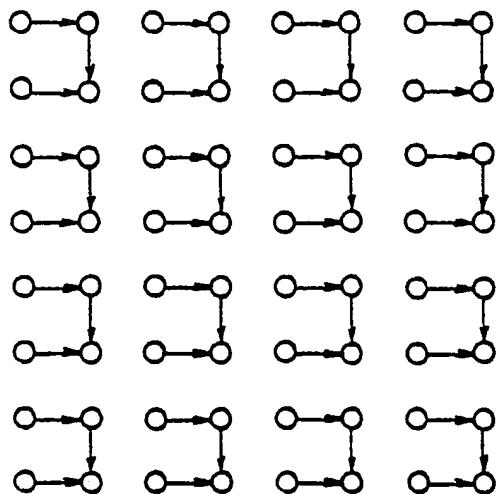


FIG. 14

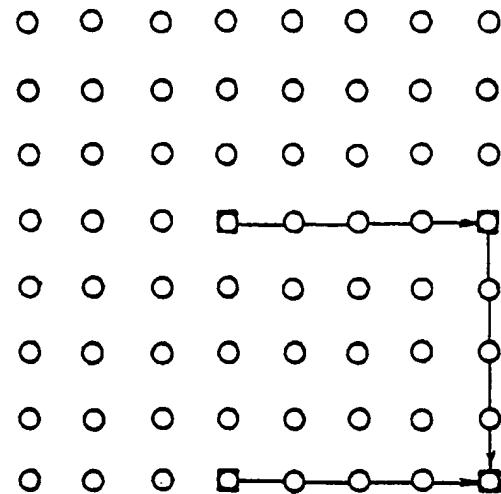


FIG. 13

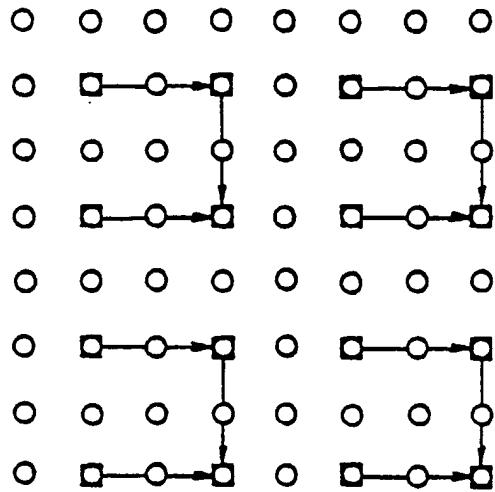


FIG. 15

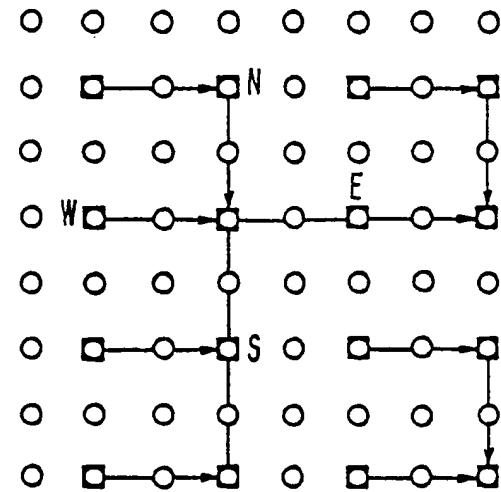


FIG. 16

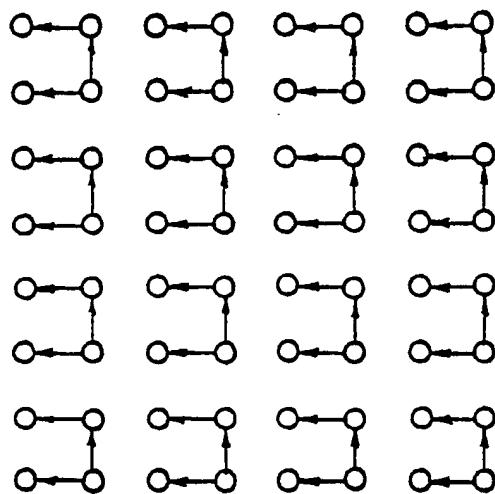


FIG. 17

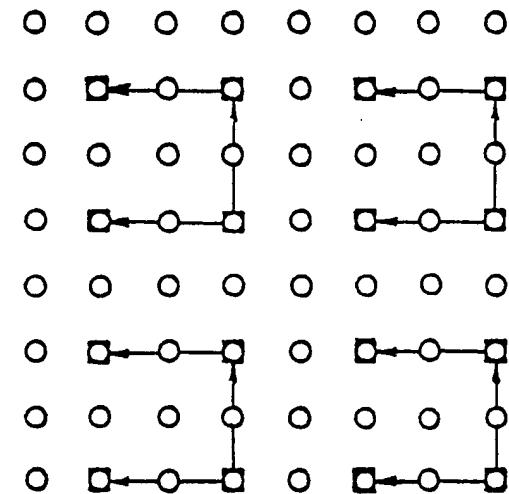


FIG. 18

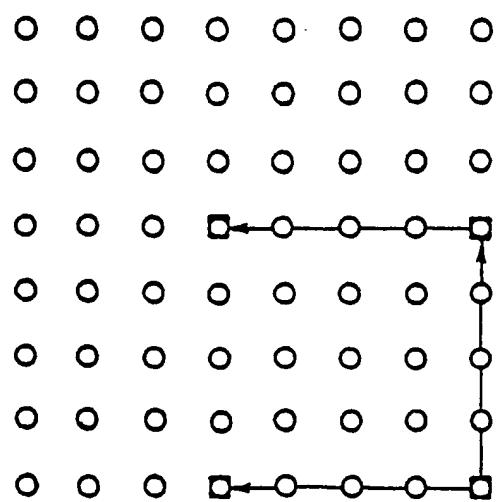


FIG. 19

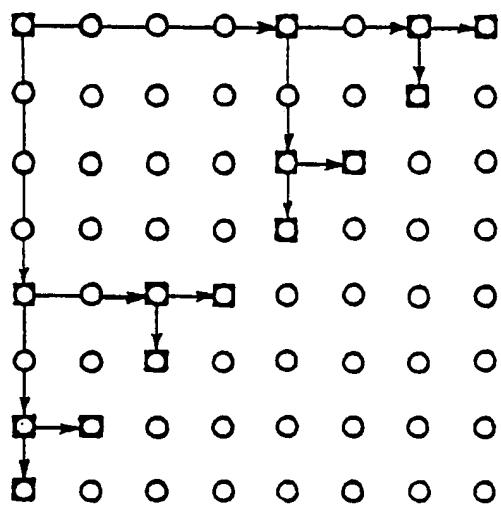


FIG. 20

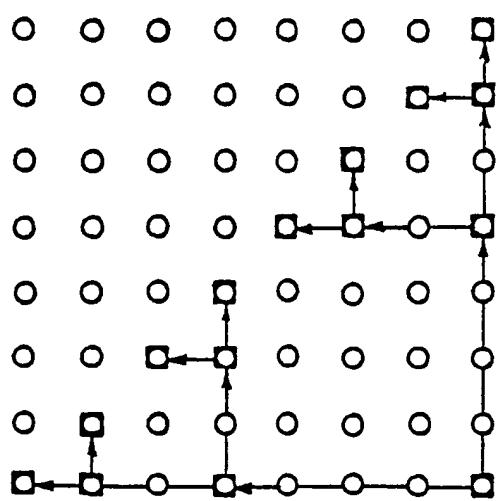


FIG. 21

